



**The Polytechnic of Wales
Politechnig Cymru**

**DIGITAL TERRAIN MODELS
FOR
RADIO PATH LOSS CALCULATIONS**

David B. Kidner

Department of Mathematics and Computing

**A thesis submitted in partial fulfilment of the requirements of the
Council for National Academic Awards (CNAA)
for the degree of Doctor of Philosophy.**

November 1991

TABLE OF CONTENTS

Table of Contents	<i>i</i>
Acknowledgements	<i>iv</i>
Declarations	<i>v</i>
Abstract	<i>vii</i>
Introduction	<i>viii</i>
Chapter One Terrain Modelling for Radio Path Profiles	1
1.1 Introduction	1
1.2 Digital Elevation Models (DEMs)	2
1.3 DEM Applications	4
1.4 Data Acquisition	7
1.5 DEM Accuracy	9
1.6 DEM Production	11
1.7 Radio Path Loss Algorithm	12
1.8 Chapter Summary	16
Chapter Two Terrain Modelling and Data Structures	18
2.1 Introduction	18
2.2 Spatial Data	20
2.3 DEM Classification	22
2.4 Topological Point DEMs	23
2.4.1 Regular Grid DEMs	23
2.4.2 Semi-Regular Point DEMs	25
2.4.3 Irregular Point DEMs	28
2.5 Mathematical Surface Models	33
2.5.1 Global Methods	34
2.5.2 Local Methods	39
2.6 Chapter Conclusions	43
Chapter Three The Regular Grid DEM	47
3.1 Introduction	47
3.2 Grid Interpolation Techniques	48
3.3 Grid Profile Interpolation	55
3.4 Radio Path Loss	57
3.5 Chapter Conclusions	60
Chapter Four Regular and Variable Sub-Sampled Grid DEMs	63
4.1 Introduction	63
4.2 Sub-Sampled Regular Grids	63
4.3 Variable Density Grid DEM	71
4.4 Chapter Conclusions	80

Chapter Five	Compression Techniques for Grid DEMs	82
5.1	Introduction	82
5.2	Differential Altitude Grids	83
5.3	Huffman Encoded Grids	90
5.4	Error Tolerant Huffman Grids	103
5.5	Sub-Sampled Huffman Grids	107
5.6	Variable Density Huffman Grids	108
5.7	Chapter Conclusions	109
Chapter Six	Polynomial Surface Patch Modelling	112
6.1	Introduction	112
6.2	Polynomial Regression	112
6.3	Polynomial Surface Patch DEM	117
6.4	Summary and Conclusions	132
6.5	Adaptive Polynomial Surface Patch Modelling	133
Chapter Seven	Surface Patch Quadtree	143
7.1	Introduction	143
7.2	Linear Quadtree	146
7.3	Mathematical Surface Patch Functions	149
7.4	Surface Patch Quadtree Implementation	151
7.5	Profile Interpolation and Radio Path Loss Performance	157
7.6	Chapter Conclusions	162
Chapter Eight	Irregular Point DEMs	164
8.1	Introduction	164
8.2	Grid Information-Rich Points	165
8.3	The Delaunay Triangulation	171
8.4	TIN Data Structures	177
8.5	TIN Terrain Results	181
8.6	TIN Profile Interpolation	188
8.7	Radio Path Loss Results	192
8.8	TIN Conclusions	195
8.9	Implicit Triangulation	196
Chapter Nine	Summary of Results and Conclusions	202
9.1	Introduction	202
9.2	Summary of Results	204
9.3	DEM Summary	216
9.4	Research Summary	221
9.5	Final Conclusions	225
Cited References		227

Appendix A	Digital Elevation Model Data	240
Appendix B	Radio Path Loss Prediction: Algorithm and Examples	247
Appendix C	Critical Surface-Specific Points	256
Appendix D	Huffman-Encoded Compression of O.S. Data Sets	267

Supplement Papers

"Digital Terrain Models for Radio Path Profiles" (1990)
by D.B. Kidner, C.B. Jones, D.G. Knight & D.H. Smith
Proceedings of the 4th International Symposium on Spatial Data Handling,
Zurich, Switzerland, July 23rd-27th, Vol.1, pp.240-249.

"Implicit Triangulations for Large Terrain Databases" (1991)
by D.B. Kidner & C.B. Jones
Proceedings of the 2nd European Conference on GIS,
Brussels, April 2nd-5th, Vol.1, pp.537-546.

"Parallel Implementation of the Delaunay Triangulation Within a
Transputer Environment" (1991)
by J.A. Ware & D.B. Kidner
Proceedings of the 2nd European Conference on GIS,
Brussels, April 2nd-5th, Vol.2, pp.1199-1208.

"Compression of Digital Elevation Models by Huffman Coding"
by D.B. Kidner & D.H. Smith
Submitted to Computers and Geosciences, August 1991.

ACKNOWLEDGEMENTS

I would like to give my unreserved thanks to my supervisors Dr. Derek Smith, Dr. Chris Jones and Dr. David Knight for their help and encouragement throughout the course of this research.

I would also like to acknowledge the support given by the Royal Signals and Radar Establishment (RSRE), Defence Research Agency, Electronics Division, Malvern. In particular I would like to thank Mr. Ray Bradbeer, Mr. Steve Bracking and Mr. Mike Partridge for many helpful discussions and other assistance during the course of this research agreement.

I am also indebted to the many members of staff who have supported my research throughout my time at the Polytechnic.

The Ordnance Survey (O.S.) data sets used in this study have been reproduced with the permission of the Controller of Her Majesty's Stationery Office (H.M.S.O.).

This work has been carried out with the support of Procurement Executive, Ministry of Defence.

Certificate of Research

This is to certify that, except where specific reference is made, the work presented in this thesis is the result of the investigation undertaken by the candidate.

Candidate *D. B. Kidner*

Director of Studies *D. H. Smith*

Declaration

This is to certify that neither this thesis or any part of it has been presented or is being currently submitted in candidature for any degree other than the degree of Doctor of Philosophy of the C.N.A.A.

Candidate *D. B. Kidmer*

DIGITAL TERRAIN MODELS FOR RADIO PATH LOSS CALCULATIONS

David B. Kidner

The Polytechnic of Wales

ABSTRACT

This work addresses the problem of digital terrain modelling for estimating radio path propagation within a mobile communication system. The ideal requirements are for a data structure which is storage efficient and computationally efficient for calculating profiles, whilst elevation errors should be constrained and radio path loss errors should be minimised. For a digital terrain model (DTM) to be considered viable as an alternative to the regular grid, it should:

- (i) produce a storage saving of at least 75% over the regular grid;
- (ii) be error constrained to a maximum absolute error of 10 metres;
- (iii) produce only a small overall average elevation error;
- (iv) preserve critical terrain characteristics such as ridges, peaks and slopes;
- (v) produce 95% of profiles to within a radio path loss error of ± 6 decibels; and
- (vi) be as computationally efficient as the regular grid.

This research focuses on the implementation of a number of prototype DTMs, including a regular grid, sub-sampled grids, variable density grids, elevation difference grids, polynomial models of fixed and variable degree, surface patch quadtrees, and triangulated irregular networks (TINs). Each of these DTMs are examined in terms of the criteria outlined above. No DTM fulfils all of these requirements. The user should identify the relative importance of each requirement before selecting a specific model. For this study, computational efficiency is identified as the criterion which can be considered the least important.

With this in mind, two original DTMs are developed. These are optimised with respect to storage and error constraints. The proposed Huffman-encoded DTM represents the deviations of a regular grid of heights from linearly predicted values as variable-length codes, whilst the Implicit TIN is a storage-efficient triangulated irregular network which reconstructs the original topology of the triangulation at the application stage. Both methods produce storage savings approaching 90% over the regular grid for the data sets tested and are suitable for parallel implementations.

INTRODUCTION

The calculation of radio path losses can be carried out most effectively if information about the terrain between the transmitter and the receiver is available. A more accurate result will be obtained if information about vegetation and building clutter is also available. The work described in this report deals with the storage and retrieval of terrain data when the only terrain information required by the radio path loss algorithm is a profile in the vertical plane containing the transmitter and the receiver.

The amount of data that must be stored is potentially very large. This research addresses the problem of selecting a method of reducing the volume of terrain data. However, in applications such as the allocation of frequencies to a mobile communication network, the number of path loss calculations necessary for both wanted and interference paths means that the time taken to retrieve a profile can be of critical importance. At the same time, methods of compacting the terrain data divide into two classes, error-free methods and methods which will introduce errors in terrain elevations, leading to additional errors in the radio path loss calculations.

It follows that in looking at methods of representing the terrain, it is necessary to look at the tradeoff between the volume of data to be stored, the profile retrieval time and the errors in the radio path loss calculation due to elevation errors.

The methods in this thesis fall into a number of main classes: regular density and variable density grid methods, polynomial patch methods, triangulated network methods, quadtree methods and methods making use of Huffman coding. Results are presented which allow these methods to be evaluated from the above three points of view.

Chapter One

Terrain Modelling for Radio Path Profiles

1.1 Introduction

Optimal positioning of radio transmitters and receivers requires the generation of numerous terrain profiles for use in path loss prediction algorithms. The rapid retrieval of profiles from very extensive digital terrain models raises the question of determining the most efficient means of storing terrain data. There are a considerable number of techniques available for representing terrain, each method having its own advantages and disadvantages. Consequently, there is a requirement for comparative studies of efficiency which can assist in discriminating between modelling techniques on the basis of application-specific criteria. For the purpose of radio path profile extraction, the major issues include minimising storage space, controlling errors in elevation, and speed of profile generation. The first two of these issues are clearly of relevance to many other applications of digital terrain models, such as, for example, intervisibility studies, visual simulation of landscape and prediction of flooding due to sea level changes.

The aim of this research is not only to perform an evaluation of some of the most commonly used surface modelling techniques, but also to identify the key features needed to give the best overall performance for calculating radio path losses in a terrain modelling system. These features have been incorporated in the design of some new models, which are specifically intended for use in the extraction of possible profiles for radio path loss estimation. Even though this application might be considered of limited use to others, terrain profile generation is a widely used application of digital terrain models. The characteristics of all the models implemented are fully explored to highlight the key advantages and most suited applications, together with an overview of their limitations and possible improvements.

In many terrain modelling applications there is a choice between selecting an existing digital terrain model and testing its performance for the specific application, or alternatively adopting a new data structure specifically aimed at the requirements of the application. These conflicting approaches have raised questions regarding a model's flexibility and efficiency. This dilemma of adopting or designing the terrain model has arisen within this programme of research. The most efficient model for calculating radio path losses cannot be determined without an investigation of the strengths and weaknesses of existing methods.

Initially, this research focused on a comprehensive study of existing models for representing

terrain. This was accomplished with regard to each terrain model's elevation errors, profile elevation errors (for both feasible and infeasible transmitter/receiver sitings), deviation in radio path loss errors, together with storage efficiency and profile generation time efficiency. These results have highlighted the advantages and disadvantages of each particular model. As a result, data structures have been developed for the specific application of profile interpolation and calculation of their corresponding radio path losses.

1.2 Digital Elevation Models (DEMs)

Surface modelling is a general term which is used to describe the process of representing a physical or artificially created surface by means of a mathematical expression. Terrain modelling is one particular category of surface modelling which deals with the specific problems of representing the surface of the Earth (Petrie and Kennie, 1987).

The concept of creating digital models of the terrain is a relatively recent development, and the introduction of the term digital terrain model or DTM is generally attributed to two American engineers working at the Photogrammetry Laboratory of the Massachusetts Institute of Technology during the late 1950s (Miller and LaFlamme, 1958). The objective of their work was to expedite highway design by digital computation based upon photogrammetrically acquired terrain data. Since then, the subject has developed considerably and is currently an area of widespread activity in cartography, surveying, geology, geophysics, civil and mining engineering and other disciplines in the earth sciences. However, the concept of a 'digital terrain model' has not been uniquely defined. Throughout the literature there are two conflicting definitions of a DTM. It can be thought of as a representation of elevations or as an integrated surface modelling package. For example,

"Digital terrain models may be defined as the numerical (or digital) and mathematical representation of a terrain by making use of adequate elevation and planimetric measurements, which are compatible in number and distribution with that terrain, so that the elevation of any point of known planimetric coordinates can be automatically interpolated with required or specified accuracy for any given application" (Ayeni, 1978).

However, Frederiksen et al (1985) state that

"the name 'digital terrain model' is misleading. A DTM is a program package consisting of routines for data storage, data retrieval, editing, interpolation and contouring. It is often a sub-system of a land information system, and is really more of an elevation information system than a model".

A similar viewpoint is taken by Heller (1986) who simply states that

"DTMs are most commonly described as data points and methods to interpolate a surface between them".

These two latter definitions have become more popular with the increasing need to incorporate DTMs within the wider framework of a geographical information system (GIS). Weibel et al (1989) and Weibel & Heller (1990) use this viewpoint as a basis for presenting a conceptual framework intended as a guideline for the development of future digital terrain modelling systems. However, since this programme of research is aimed at deducing the most suitable model for predicting radio path losses, a general surface modelling package is of limited value. In general, there will always be many users who require a DTM for a specific application or a limited number of operations. Hence in this study, the definition of a DTM is analogous to that of Ayeni (1978) above.

The term digital elevation model (DEM) is also commonly used to mean DTM, but because the term 'terrain' often implies attributes of landscape other than the altitude of the land-surface, including derived data about the terrain such as slope, aspect, visibility, etc., the term DEM is more commonly preferred for models containing only elevation data (Petrie and Kennie, 1987). The decision whether to use the term DEM or DTM, can therefore be left to the individual, but throughout this thesis, the term DEM is used. The terms digital height model (DHM) and digital ground model (DGM) are also occasionally used in the literature.

Whatever the format of the data, DEMs represent a convenient way of storing elevation data, and of making such information available to applications programs. As a result, the utilisation of DEMs is becoming increasingly important, and their generation has become a major area of production for many national cartographic organisations. Many possible data structures can be adopted to represent the DEM. These can be broadly classified into models which structure the points into some specific order, taking into account their spatial relationships, and models which fit mathematical functions to the elevation data.

The most popular DEM data structure is the regular grid, in which points are stored at regular intervals in both the X and Y directions, thus forming a regular lattice of points. Each data point is stored as an element in a two-dimensional matrix or array, such that the fixed grid spacing of points allows the search for a point to be implied directly from its coordinates. This relationship between coordinates and matrix position means that the X and Y coordinates of each point need not be stored in the data structure, as long as the coordinates of the origin and the grid spacing are known. However, this type of DEM has inherent inflexibility, since the structure is not adaptive to the variability of the terrain. As a result, the effect of modelling the surface at the same resolution throughout will create excessive storage requirements or data redundancy.

Despite the dramatic fall in computer costs in recent years, the storage overheads of a terrain database based on the regular grid DEM are impractical and beyond the means of some users.

The application for which this research is primarily intended is for the siting of radio transmitters and receivers in a mobile communications network. The cost of 'ruggedised' hardware, capable of withstanding the effects of mobilisation over a wide variety and range of terrain is considerably higher than for other users, thus highlighting the need for storage efficiency.

The specific application of calculating radio path losses requires an estimate of terrain profiles between the possible transmitter and receiver locations. These profiles need to be interpolated from the DEM, and in some instances it is necessary for this profile extraction to be very fast. When using the regular grid DEM, the profile is often calculated using a simple method, such as linear or bilinear interpolation, which are both very time-efficient. This is because the neighbouring points involved in each interpolation can be directly accessed in the data structure from the coordinates of each profile point. However, despite this time-efficiency for retrieval operations, the overheads of storage requirements may outweigh the benefits of speed, or more specifically the data structure may require more space than the computer's storage capacity will allow.

This highlights one of the major problems faced during this research and with DEMs in general. The excessive storage costs of most DEMs are overlooked in favour of their flexibility at the application stage and good time efficiency in general. Thus the application-specific user will have to be content with either using an existing DEM which might not be best-suited to his requirements (particularly storage costs), or alternatively developing a new DEM. The latter approach also incurs problems, such as model inflexibility and development costs. However, this decision is more usually influenced by the degree to which storage efficiency can be improved compared to the decrease in time efficiency. Savings in storage are usually achieved at a cost of increasing search time within the model's data structure. Therefore the main aim of this research is to identify possible alternatives to the regular grid DEM as a storage-efficient means of representing terrain, yet which will be time-efficient for interpolating profiles and calculating radio-path losses.

1.3 DEM Applications

The mathematical operations involved in the application of DEM height data have been summarised by Doyle (1978). Other authors, including Collins (1981), Yoeli (1983a), Burrough (1986), Catlow (1986), and Petrie & Kennie (1987) have categorised the applications and benefits of DEMs. These applications can vary quite considerably and include topographical, geological, geophysical, hydrographical and bathymetrical mapping, civil engineering, mining engineering, simulation and terrain visualisation and military engineering. More

specifically, these applications include :

- (i) Determination of contour lines;
- (ii) Generation of profiles;
- (iii) Earthwork calculations (mainly for road design);
- (iv) Cross-country visibility analysis;
- (v) Slope maps, aspect maps and slope profiles;
- (vi) Shaded relief maps;
- (vii) Terrain simulation;
- (viii) Construction of isometric, perspective and panoramic plots;
- (ix) Statistical analysis and comparison of different terrain;
- (x) Seabed maps;
- (xi) Geological maps of underground surfaces;
- (xii) Energy exploration;
- (xiii) Navigation control systems;
- (xiv) 3-D displays of landforms;
- (xv) Communication network siting.

Most of these applications are available in the form of software packages. Many are also incorporated in some of the geographical information systems (GIS) that are on the market. With this new interest expanding rapidly, it seems likely that terrain modelling methods will continue to develop and expand into other areas of activity. The development of GIS has led to a greater demand in DEMs for visualisation applications (L'Eplattenier & Sieber, 1986; Sieber, 1986; McCullagh, 1987, 1988; Kennie & McLaren, 1988; Weibel & Herzog, 1989; and Clarke (Ch.11), 1990). However, as Kennie & McLaren (1988) point out:

"visualization techniques have released the world from its traditional two dimensional approaches to display and, in so doing, have highlighted the three dimensional deficiencies in our sources of data in terms of availability and accuracy".

This issue is addressed in Section 1.5.

GIS are a powerful set of tools for collecting, storing, retrieving , transforming, and displaying spatial data from the real world for a particular set of purposes (Burrough, 1986). They represent a rapidly developing field lying at the intersection of many disciplines and are of interest to a wide and increasing number of users. Systems are being developed at scales from the cadastral to the global and for a wide variety of purposes, yet they have many features in common and face many similar problems (Coppock and Anderson, 1987). The structure of these systems vary considerably, since they adapt to the specific requirements and constraints imposed by the data (Little, 1978).

Some of the applications outlined above need other parameters or information as well as elevation in the model (for example, geological maps), so the DEM is only a part of that model. Computer programs which employ such models can be broadly classified as GIS. Little (1978) broadly classifies GIS as those which assume the value of a model variable to be constant within each of a set of regions (coverage systems), and those which assume continuously varying data (digital terrain models). Geographic information is data which can be related to specific locations on the Earth. It covers an enormous range, including the distribution of natural resources, the incidence of pollutants, descriptions of infrastructure such as buildings, utility and transport services, patterns of land use and the health, wealth, employment, housing and voting habits of people (Chorley, 1987).

The development of GIS has been parallel to the development of DEMs, starting with the technological advances in computation, cartography and photogrammetry in the 1940s and 1950s. It was recognised that GIS could manipulate and analyse data to provide output which could be used as part of a decision-making process. There has been a rapid increase in the number of GIS, as a result of both advances in computer technology and increases in the availability of spatially-referenced data in digital form. Until recently, the computer systems employed to handle geographic information were too expensive and cumbersome for most users. Two developments have rapidly changed this. The first is the dramatic fall in computer costs, processing costs are falling by a factor of one hundred every ten years, with the expectation of similar changes in the future. In addition, the growth in packaged software has significantly reduced system development costs. Along with this has been the increasing ease of using computer systems (Chorley, 1987).

The function of an information system is to improve a user's ability to make decisions in research, planning and management. An information system involves a chain of steps from the observation and collection of data through their analysis to their use in some decision making process. Computer-based GIS may be viewed as having five component sub-systems including data encoding and input processing, data management, retrieval, manipulation and display.

Thus GIS encompass the software and hardware necessary for users to plan and make decisions quickly. The DEM will play an important role in this configuration, since the attributes of the terrain and their spatial relationship, will more often than not, have to be included in the database. There is a growing tendency for DEMs to be designed so that their data structure can if possible include attributes and relationships. Peucker and Chrisman (1975), recognised this problem, outlining types of cartographic and geographic data structures which have information incorporated as well as elevation.

The application of GIS is still at an early stage of development, since in most application areas

there are signs of significant future growth in GIS (Chorley, 1987). This is verified somewhat by the North American and UK suppliers of hardware and software, who predict a steep growth in demand for their products over the next few years. It is important therefore, that efficient data structures (including DEMs), for modelling these information systems continue to develop in the future. Weibel & Heller (1990) believe that the meaningful use and exploitation of terrain modelling within GIS is rather limited, due to research on rather narrow problems and applications. Weibel et al (1989) and Weibel & Heller (1990) identify some of these shortcomings and present an approach or 'conceptual framework' for future terrain modelling systems.

A current issue in GIS research which is relevant to the maintenance of a large terrain database is the integration of elevation data into databases at international, continental or global scales. Two such hierarchical data structures which address this problem are the geodesic elevation model (GEM) and the Quaternary Triangular Mesh (QTM) (Douglas, 1984, 1989, 1991).

1.4 Data Acquisition

Before the advent of DEMs, most applications were implemented cartometrically (through measurements on the contour map). Contours in the cartographic sense are the projection onto the map plane of lines connecting points of equal elevation on a three-dimensional surface, and are considered to be a rigid measuring scale superimposed on the relief (Yoeli, 1975, 1983a).

In DEMs, as with contours, the continuous surface is approximated by discrete values, but there is a fundamental difference in that while the quantity of terrain information on a contour map depends on its scale, the point density of a DEM is optional. Its degree of approximation to the true surface can be regulated at will, provided the necessary source material is available. This accuracy depends on the source from which the spatial coordinates are measured, the density of the height points and their relative geometric situation. The denser the net, the more accurate the results, and the greater the cost. For most DEMs there are two stages of development :

- (i) Initial stage comprising the input information (at a relatively low density),
- (ii) Interpolation stage based on these input points, to a dense regular grid.

The acquisition of the basic terrain data required for a DEM is often a major task, which has not been made any easier by the lack of a national topographic database. DEMs originate from three main sources :

- (a) Terrestrial or ground surveys,
- (b) Photogrammetric surveys,
- (c) Topographic contour maps.

The most frequent form of terrestrial DEMs consist of points surveyed tacheometrically. As far as their geometrical disposition is concerned, they are completely randomly distributed. Geomorphologically however, they may be carefully placed at characteristic points of the terrain, ie. places of slope-change or along the structure lines of the relief such as valleys or ridges. Other ground surveys include engineering surveys and area levelling.

The principle of obtaining photogrammetric measurements of the terrain is to create an exact three-dimensional stereoscopic model from aerial photographs, by using a stereo-plotting machine (Petrie, 1987a). The photogrammetrist then measures the stereo-model very accurately instead of carrying out measurements in the field using surveying instruments. The savings in time and cost of doing so are great. The elevations may be obtained in several sampling patterns, although systematic or grid-based sampling is the most common. A pattern of spot heights may be measured in a regular geometric pattern (square, rectangular, triangular, hexagonal). This grid can be preset to a specific interval, which has the consequence that the finer but perhaps significant terrain features will not be measured specifically. Other techniques include random sampling, composite sampling (grid measurements that are supplemented by significant points in the terrain), measuring contours, and vertical profiling or 'dropped lines' (Yoeli, 1975; Pratt, 1979; Petrie, 1987a).

Contour maps represent an important source of data for DEMs, since much of the earth's surface has been mapped in this way. Another approach to data acquisition deals with the measurement of contour lines, so that they are represented by suitably structured strings of digital coordinate data. Even if the original maps have been measured photogrammetrically, the contour lines will be considerably less accurate than the measured spot heights produced by field survey or photogrammetric procedures. There are two main methods by which digitising is implemented, namely line following and raster scanning. These measurements may be executed either manually, or automatically using a suitably designed machine. Some other methods include covering the map with a regular mesh and interpolating the heights of the mesh points between the contour heights, or even scanning the map on arbitrary lines (Yoeli, 1975; Petrie, 1987b).

Surprisingly, little research has been implemented to assess the feasibility of satellite data for digital elevation models. Theodossiou & Dowman (1990) discuss some of the research in this area, with particular respect to SPOT stereodata. Their conclusions suggest that satellite imagery has a potential for providing data for topographic mapping. However, there are significant systematic errors due to a number of reasons, many of which could be overcome by future research, including more detailed qualitative assessments of results.

Once the data has been acquired, it may be converted to a convenient format or data structure,

which is usually a grid. From the methods outlined, it can be seen that the sampling methods vary considerably, with some methods having no obvious spatial relationships. The most generally accepted format for storing data is in the form of a regular two-dimensional grid, so for most of the methods some form of data interpolation is necessary. Petrie (1987c) distinguishes these methods as:

- (a) **Pointwise Methods** - Independent determination of different functional parameters and height values for each and every grid node being interpolated.
- (b) **Global Methods** - A single complex three-dimensional surface fitted to the complete data set, with the interpolation of terrain heights at all the required nodes on the regular grid.
- (c) **Patchwise Methods** - Three-dimensional surfaces or patches are established, from which the elevations of grid points lying within each patch can be interpolated.

Schut (1976), categorises interpolation methods into six groups, depending on the sampling pattern used for acquisition. Some of the most popular interpolation methods that are used in digital mapping are discussed by Shepard (1969), Crain (1970), Rhind (1975), Schumaker (1976), Pratt (1979), McCullagh (1981), Lam (1983), Davis (1986) and Yoeli (1975, 1986). Makorovic (1976), also discusses interpolation in detail, highlighting the steps involved. These consist of selecting the reference or required data points by means of a search, or patchwise partitioning, and converting this data by interpolation, which may be pointwise, patchwise, or combined.

An additional constraint which will affect the performance of interpolation, is the choice of the grid size itself. It is necessary to consider the likely requirements of grid size related to the number of data points in the area of concern (McCullagh, 1987, 1988). McCullagh states that information theory leads to the conclusion that the number of grid interpolations should be roughly equivalent to the number of data points, but accuracy requirements preclude the use of such small grids. 'Grid size determination for a data set is at best difficult, and at worst leads to significant errors in over-generalising the surface and removing data fluctuations' (McCullagh, 1988). As a result, the process does not honour the data points unless they occur exactly at the grid locations.

1.5 DEM Accuracy

The DEM in whatever form should provide an accurate representation of the real surface. It is difficult, however, to test how accurately any digital model actually captures the real world, since there is no independent model to test the digital model against (Carter, 1988). Accuracy is a prime consideration of the DEM user, yet has received little attention, particularly with respect to its effect on applications. Dutton (1984) states:

"In general however, the overall fidelity of terrain data is difficult to assess without detailed information of how the source data were collected, edited, reproduced and (sometimes) interpolated. As such, they will usually contain errors which will persist without notice, but unfortunately not without consequence, indefinitely into the future".

Ackermann (1978) carried out experimental contouring comparisons between digital, photogrammetric and surveyed data, but points out that no standard exists, by which comparisons of various DEMs can be made. This leads to the conclusion that the DEM user decides if it is of suitable accuracy by either intuition or trial and error. Ackermann does show that the resulting accuracy depends to a great extent on the method of data acquisition. This leads us to consider which of the three main data acquisition methods (terrestrial surveys, photogrammetric surveys or digitised contour maps) is the most accurate.

Digitised contours will never produce the same accuracy as the other methods, since the accuracy of such contours is only one-third of that of directly measured spot heights. Furthermore, interpolation to regular grid DEMs will introduce further errors. For photogrammetrically measured terrain data, the expected height accuracies of elevations expressed as root mean square error (RMSE) will lie in the range ± 0.1 to 3.0 metres over the possible range of flying heights up to 15 kilometres (Petrie, 1987a). Terrestrial surveys will give better results, but are not practical for large areas and are limited in use to small area site planning and design. Kennie & McLaren (1988) tabulate the relative accuracies of each of these methods of data acquisition.

Ackermann's results (1978) confirm that the ideal distribution is not a uniform density of points, but one which is adapted to the characteristic features of the terrain. A regular grid DEM, even if acquired photogrammetrically would have to be very fine to capture all of the features. A coarser grid is likely to miss critical points such as peaks and pits - features whose heights are difficult to interpolate accurately in any case. Also, since a regular grid DEM may have been interpolated, accuracy will again be affected. Hannah (1981) highlights algorithms to detect and correct errors, incurred from photogrammetric data acquisition. Ley (1986) has also addressed the problem of accuracy standards and discusses potential ways of measuring this. The spatial frequency of the sample elevations and the precision of the data are two other factors that must be considered. To capture the finest details in the land surface, a large number of data values must be digitised, so the user must be prepared to pay the high price of creating a detailed database (Carter, 1988).

Irregularly sampled DEMs attempt to overcome these accuracy problems, but at a cost of creating additional problems, such as data storage and accessibility. Accuracy is therefore directly related to the method of data acquisition and thus cost. In many instances, the actual terrain variability is of secondary importance. To many users, cost is the most important factor,

which may cause the accuracy of the DEM to be overlooked. Makarovic (1976) has summarised the relationship between DEM accuracy and data acquisition as:

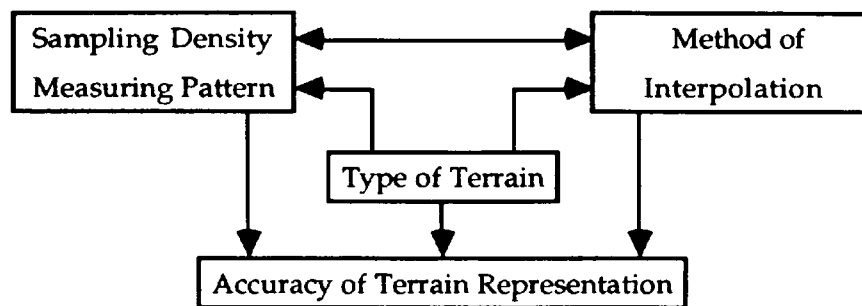


Figure 1.1 - Factors Influencing the Performance of a Digital Elevation Model (Makarovic, 1976)

Whilst the existence of errors in data sets and DEMs has been widely acknowledged, the effect that they have on applications has received little attention. Fisher (1991) provides a simulation and evaluation of grid DEM error propagation for viewshed (line-of-sight) operations. The results suggest that even small RMS errors may seriously affect the calculated viewshed. Therefore, for any DEM, it should be expected that errors are very likely to occur in line-of-sight calculations for operations such as radio path loss estimation. If this is true for the regular grid DEM, even more errors are likely to arise in alternative DEMs which approximate the original data sets.

This study will therefore help to estimate the radio path loss errors associated with each terrain model. Furthermore, the effect of elevation error on radio path loss prediction can be estimated. Since the location and reason for these elevation errors are known in the alternative DEMs, the effect of errors in particular terrain features can be determined.

1.6 DEM Production

Photogrammetric techniques of data acquisition can be considered the most effective data acquisition method with respect to accuracy. Highly developed systems are extensively used by the main government mapping agencies in N. America. The density of the terrain elevation data acquired with such systems is extremely high, with some 500,000 to 750,000 elevation points being obtained from a single stereo-model. Unfortunately, the cost of buying and running such automated systems is high and can normally only be justified where there is a large programme of digitising to be carried out, as with national mapping agencies, or where the overriding requirement is for rapid measurement and cost is a subsidiary factor (Petrie, 1987a).

The United States Geological Survey (U.S.G.S.), the main government mapping agency, produces DEMs as a regular array of elevations stored every 30 metres (Allder et al., 1982;

U.S.G.S., 1990). Each DEM contains between 138,000 and 195,000 elevation points at a scale of 1:24,000. The Defence Mapping Agency (D.M.A.) in America, produces 1:250,000 scale DEM data which is derived entirely by digitising the contours, ridges, and channels shown on existing 1:250,000 scale maps, and interpolating at intervals of three arc-seconds. The U.S.G.S. DEMs have a root mean square error (RMSE) of 7 metres, whilst the D.M.A. DEMs have a significantly lower accuracy, especially in areas of steep terrain.

In the U.K., the Ordnance Survey (O.S.) has been very active in digital mapping since the early 1970s, but does not produce any photo-map series from which terrain elevation information would be gathered as a by-product of photogrammetric scanning. Only recently has interest in digital data caused the O.S. to reconsider the nature of digital data as a product in its own right, and not purely as a technique of map production (Sowton, 1989). In 1987 the Committee of Enquiry into GIS (Chorley, 1987) added considerably to the already growing pressure on the O.S. to increase the output of digital map data, by collaborating with its major customers in the funding and acceleration of the digital conversion programme.

However, the needs of the military have led to an active and extensive programme of digitising existing cartographic material of the U.K. at the 1:250,000 scale to generate digital elevation data. From this data various program packages have been used to generate regular grid DEMs, such as the Digital Land Mass Simulation (DLMS), which has been interpolated for the whole country. This work has been carried out by the Mapping and Charting Establishment (M.C.E.) of the Ministry of Defence (Petrie, 1987b). This data has a limited resolution and accuracy, but has proven to be suitable for its intended purpose of radar and aircraft simulation, and visibility studies. A further development has been the initiation of a project by M.C.E. (due for completion by 1992) to generate regular grid DEMs at an improved resolution and accuracy for the whole of the U.K., via digitising of the existing contours on the 1:50,000 scale O.S. map series (Petrie, 1987b, Morris & Flavin, 1990). At present, the U.K. is sadly lacking behind other European countries, including Sweden and West Germany (Petrie, 1987a, 1987b), in terms of the availability of good, accurate digital terrain data.

1.7 Radio Path Loss Algorithm

"Radio wave propagation includes everything that can happen to the energy radiated from a transmitting antenna during its journey to the receiving antenna. It includes the radiating properties of both antennas, such as gain, directivity and polarisation; it includes free space attenuation of the wave with distance, and encompasses such factors as refraction, interference, diffraction, absorption and scattering. Propagation is therefore dependent upon the properties of all transmission and boundary media" (Reed & Russell, 1966, p.1).

The entire radio frequency spectrum is divided into adjacent bands, each of which has been

given a distinct name. In this study the frequency range used for the radio path loss algorithm is 200 megacycles per second (Mc/s) or megaHertz (MHz) to 1800 MHz at signal ranges of up to 28 kilometres. This includes the VHF (very high frequency) bandwidth of 30 - 300 MHz (wavelength of 10 metres to 1 metre) and the UHF (ultra high frequency) bandwidth of 300 - 3000 MHz (wavelength of 1 metre to 10 centimetres). At these frequencies, the energy is travelling in the lower region of the atmosphere, the troposphere, in which most of the effects of weather take place (Matthews, 1965). One of the largest user groups of mobile radio and radio links in the U.K. are the energy producing industries. These mobile radio services operate in the VHF band, whilst the radio links use the bands 450/470 MHz, 1500 MHz and 7000 MHz (Dadson, 1979).

The planning and design of radio services require that detailed information is available concerning the terrain in areas where radio coverage is required (Dadson, 1979). The Joint Radio Committee (JRC) set up a project in 1967 to assess the feasibility of using digital elevation models for the calculation of radio network area coverage. Edwards & Durkin (1969) first reported the benefits of using DEMs in a VHF mobile radio network.

The algorithm used in this study assumes that no external factors, other than the terrain elevations, will affect the propagation. Radio path loss is calculated solely from the terrain data structure, by an estimation of the terrain profile between the transmitter and receiver. Other physical influences on radio path loss, such as clutter from vegetation or buildings could be easily incorporated within the DEM, provided such information is available. In the simplest case, data points within the DEM could be flagged for a corresponding clutter element. Akeret (1989) reports that a one-bit flag is used to indicate the presence of vegetation or an object of at least 14 metres in the mobile subscriber equipment (MSE) system for path profiling of VHF radio networks.

Methods of calculating the attenuation to be expected over transmission paths in point-to-point radio links have been well developed for both very high and ultra high frequencies. A review and comparison of a number of these methods in the VHF range has been made by Grosskopf (1987), using topographical data from a DEM. These are classified into empirical, semi-theoretical and theoretical methods. Baker et al (1983) compare the results for a number of radio path loss prediction methods at VHF and UHF, particularly for base-to-mobile applications within London. Other simple methods for deriving calculations for point-to-point links and mobile radio networks are described by Edwards & Durkin (1969), Dadson (1979) and Symmons (1982).

The path loss algorithm used in this study is based on a similar principle to that of Edwards & Durkin (1969) and Meeks (1983), in that it is a knife-edge diffraction model. The algorithm is

described in greater detail by Jones & Knight (1987, 1988). The maximum number of obstructions for which diffraction losses are calculated is three. When radio waves encounter an obstructing object, such as a peak, some of the energy is diffracted at the edges of the object and becomes bent around that edge. This reduces the shadowing effect of objects which are opaque to radio waves, as diffraction fills part of the shadow area with some energy of the wave (Reed & Russell, 1966, p.9).

The ground profile along the radial from the transmitter to the receiver is first reconstructed from the DEM, yielding profile distance and elevation matrices (d_i, h_i for $i=1,2,\dots,n$ points). The points may be interpolated at constant or irregular intervals. The elevations of the profile are corrected for the effect of the curvature of the earth. The algorithm assumes that there are no losses due to clutter or polarisation. Instead, the losses are calculated as the sum of the free space loss, the reflection or siting losses, and the losses for up to three diffraction edges (Jones & Knight, 1987). If the ground profile interrupts the transmission path, the receiver is said to be 'in shadow'. Since the critical features of the profile will generally be at these diffraction edges, they have a significant effect on the performance of the attenuation calculations. It is therefore important that these features can be retrieved from the DEM to a sufficient accuracy. These obstructions are located by using the following algorithm :

- (1) From the transmitter, calculate the angles α_i between the radial and each elevation. If $\alpha_i < 0$ for all points, the path is unobstructed or line of sight. If $\alpha_i > 0$ then the maximum α_i locates the first obstruction, (p) in Figure 1.2.

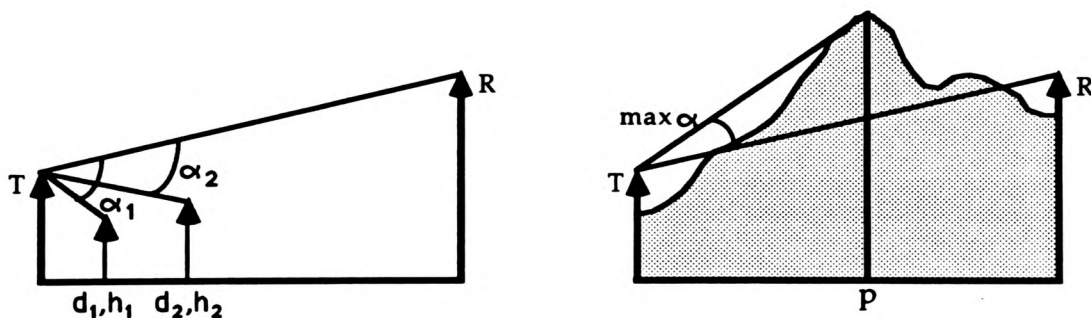


Figure 1.2 - Locating an Obstruction or 'Knife-Edge' Diffraction.

- (2) Procedure (1) is repeated from the receiver. If the located obstruction (q) is the same as p, then there is only one obstruction or diffraction edge.
- (3) If $q \neq p$, then the procedure is repeated from p to q. If no obstructions are located, then there are only two diffraction edges. Otherwise, the third obstruction (r) is recorded. The possible instances of diffraction edges are illustrated in Figure 1.3.

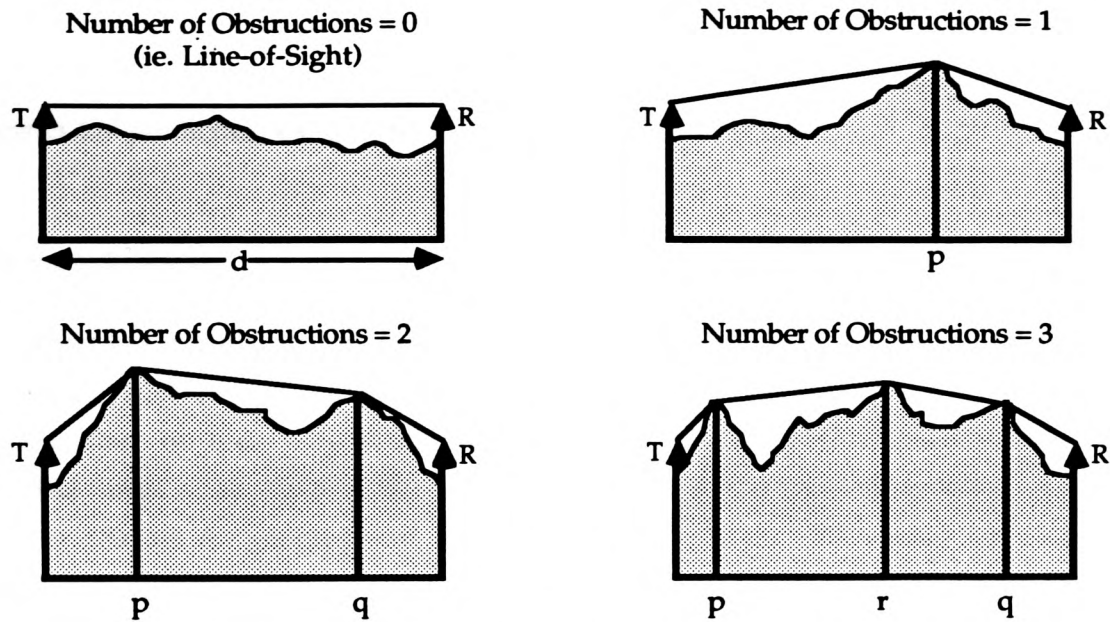


Figure 1.3 - Illustration of the Four Possible Diffraction-Edge Paths.

The obstructions or diffraction edges are regarded as knife-edges. However, problems may arise with 'rounded' hills, since the algorithm could yield more than one obstruction (see Figure 1.4).

In these instances, the peak is located by considering only those α_i at which $P_i A_i \geq P_{i+n} A_{i+n}$, where n can be arbitrarily chosen depending on distance, to exclude small, insignificant local fluctuations in the terrain. Hence, in Figure 1.4, the single obstruction would be located at P_j . Another problem that may arise in the original algorithm is the identification of two knife-edges on the same hill or ridge. To overcome this, the search for the next diffraction edge is limited to exclude part of the first hill or ridge. This is accomplished by introducing a specified distance tolerance from the knife-edge, which Meeks (1983) terms the 'characteristic length'. A typical value for this may be one kilometre. These two amendments to the algorithm of Jones & Knight (1987, 1988) ensure a much greater consistency of results, since more distinct knife-edges are determined.

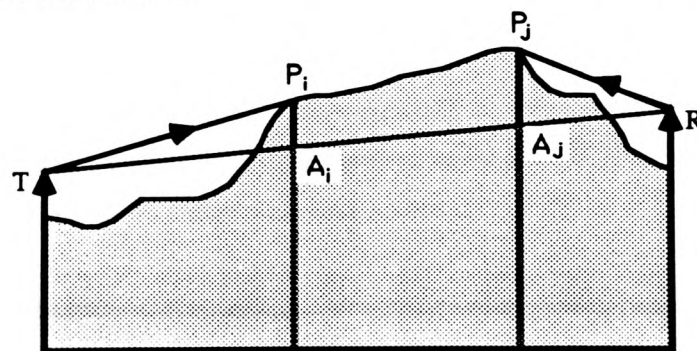


Figure 1.4 - Identification of Diffraction Edges on Rounded Hills.

For a detailed description of the algorithm, including the formulae used, the reader should refer to the work of Jones & Knight (1987). A further description of the algorithm, together

with some illustrated examples for profiles of the types shown in Figure 1.3 are presented in Appendix B - Radio Path Loss Prediction: Algorithm and Examples. The algorithm is also illustrated for predicting the effective service area of a transmitter.

1.8 Chapter Summary

This chapter has introduced some of the fundamental issues concerning DEMs, with respect to wider aspects such as application, data acquisition and accuracy. These features should not be neglected, since by their very nature they will determine the performance efficiency of the DEM. The range of applications for DEMs is immense, and is continuing to expand with the growth in demand for geographical information systems (GIS). With this continued growth, the need for efficient spatial data structures (DEMs) will be increasingly important.

The aim of this research was to design an extensive digital terrain model for use in profile generation for calculating radio path losses between possible transmitter and receiver locations, as part of a mobile communications network. As well as being an accurate representation of the terrain, this model should ideally be both storage efficient and time efficient for searching and accessing the data. The critical features of the terrain need to be incorporated, such that accurate elevation profiles can be interpolated. For the design of this data structure, an evaluation of existing methods of representing terrain was required, such that certain factors and features could be identified, including :

- (i) Advantages of various terrain data structures;
- (ii) Advantages of surface modelling techniques;
- (iii) Characteristics of critical terrain features;
- (iv) Accurate forms of representing and storing terrain;
- (v) Storage efficient models;
- (vi) Time efficient models (ie. access and retrieval of data for profile generation);
- (vii) Error-constrained models;
- (viii) Modelling errors (ie. where and why they occur);
- (ix) Radio path loss errors (ie. where and why they occur);
- (x) Effect of elevation error on radio path loss error.

This comprehensive evaluation was necessary to determine the requirements of the new model. However, the degree of importance to which each feature contributes, and thus its relative weighting in this structure, needed more careful consideration.

The most common approach to storing terrain data is in the form of a grid of elevations sampled at constant intervals. Despite its flexibility and efficiency for a wide range of applications,

the regular grid has inherent data redundancy and is thus too inefficient in terms of storage for some applications. Since the availability of accurate, dense, digital terrain data is likely to increase significantly in the future, especially with the parallel growth of GIS, the need for storage-efficient data structures is likely to be a key issue.

Methods of data acquisition will dictate the nature of the data structure used, as well as giving an indication to the range of errors that can be expected. For large DEMs, ground surveys are too impractical to be considered viable, since they are a costly, slow, labour-intensive method of obtaining data. Digitising of contour maps is far from ideal, due to accuracy considerations. Since most DEMs are based on the regular grid, contours are usually interpolated into this form, creating even more errors. Photogrammetric measurements of the terrain, sampled in a regular grid format, may be considered the best data acquisition technique. Accuracy is of a high standard, but the cost is beyond the means of most users and the problems associated with the regular grid are still prevalent. Satellite imagery is increasingly being used to obtain data, but there are still doubts over its accuracy, since some large errors may occur (Theodossiou & Dowman, 1990).

The O.S. is currently producing regular grid DEMs by digitising contour maps at the 1:50,000 scale and interpolating onto a regular 50 metre grid. The task is very time consuming, and the whole of Britain is still not available, but the effort is being made to remedy the shortage of adequate DEM data. However, the accuracy and grid spacing of this data will be far from ideal for some users. Since the application of GIS is still at an early stage, it is imperative that the demands for such systems are not hindered by the shortage of data.

For this research the O.S. 50 metre regular grids are used as the 'real world' models to which all other terrain models are compared. This assumes that the data are an exact representation of the terrain, and from which all other data models are derived. This is the best way of ensuring consistency for model comparison and is the most cost-effective way of acquiring data for this research. The assumption that the source data are error-free is necessary to make in order to address the problems considered in this work. These data sets are described in Appendix A.

Chapter Two

Terrain Modelling and Data Structures

2.1 Introduction

Although there are different and sometimes conflicting ways of categorising terrain data structures, the simplest and perhaps most favoured approach is on the basis of structures which store points (or lines) and topological relationships, and structures which store approximating or interpolating mathematical functions. In selecting a data structure for an application, a dilemma is encountered: the choice of an existing DEM or alternatively adopting a new data structure. An extensive literature review was undertaken, to examine the various data structures commonly used in terrain modelling, and to evaluate their advantages and disadvantages with particular respect to the application of generating terrain profiles for estimating radio path loss.

Data models may broadly be defined as specific collections of facts, entities or objects (data), together with the relations between them. They may also include a collection of operators and a collection of general integrity rules (Peuquet, 1984). The purpose of such a model is to provide a formal means of representing information, and a formal means of manipulating such a representation (Date, 1983, pp.182). The data structure however, does not take into account the nature of the problems to be solved, the storage and retrieval of data, and influences such as the characteristics of the computer system (Bouillé, 1978). Hence, the data model is required at a more general, higher abstraction level, which considers the topological aspects before those concerning the metrics (Bouillé, 1978). Peuquet (1984) states :

"... this (the data model) is a human conceptualization and tends to be tailored to a given application; different users and different applications are likely to have different models to represent the same phenomenon. As the word 'model' implies, the most basic characteristic of a data model is that it is an abstraction of reality. Each data model represents reality with a varying level of completeness."

"Many data model designers realize that in order to determine how a collection of data is to be ultimately represented in digital form, the data need to be viewed at a number of levels. These levels progress from reality, through the abstract, user-oriented information structure, to the concrete machine-oriented storage structure."

Peuquet (1984) distinguishes four levels of data abstraction :

- | | | |
|-------------------|---|--|
| <i>"Reality</i> | - | <i>the phenomenon as it actually exists, including all aspects which may or may not be perceived by individuals;</i> |
| <i>Data Model</i> | - | <i>an abstraction of the real world which incorporates only those properties thought to be relevant to the application(s) at hand, usually a human conceptualization of reality;</i> |

Data Structure - a representation of the data model often expressed in terms of diagrams, lists and arrays designed to reflect the recording of the data in computer code;

File Structure - the representation of the data in storage hardware."

In particular, the last three views or levels correspond to the major steps involved in database design and consideration. A data structure is built upon the data model, which details the arrangement of the data elements and the relationships between these objects, expressed explicitly or implicitly. Explicit relationships are written into the data structure as data elements themselves, whilst implicit relationships are indicated by the relative position of the individual data elements (Peuquet, 1984).

There exist two contrasting approaches to data model design. The first approach, termed 'phenomenon-based' design, attempts to model all identifiable entities and their relationships, such that it becomes a near complete representation of reality, and hence very complex. Alternatively, the model could be designed primarily for its intended use and exclude any entities and relationships not relevant to that use (Figure 2.1).

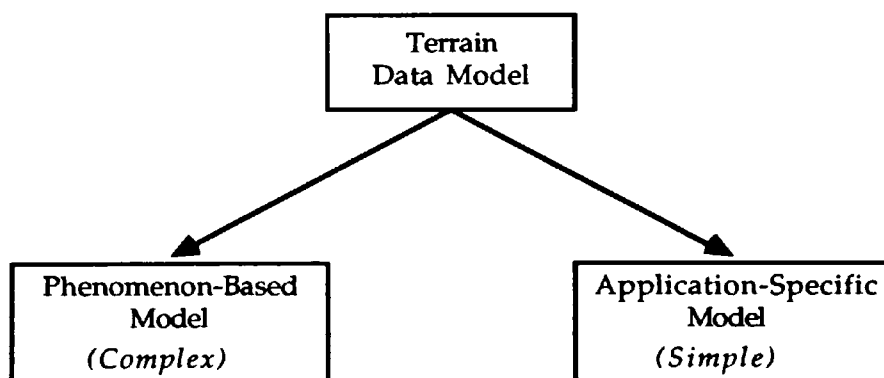


Figure 2.1 - Spatial Data Model Design.

The more perfectly a model represents reality, (ie. phenomenon-based), the more robust and flexible that model will be in application. However, the more precisely the model fits a single application, the more efficient it will tend to be in storage space and ease of use. The selection or design of a data model should ideally be based on a 'tradeoff' between these two different approaches, ie. the nature of the phenomenon that the data represents and the specific manipulation processes which will be required to be performed on the data (Mark, 1978a). Once chosen and implemented, the data model will often be difficult or expensive to modify, and if poorly designed, may unduly restrict the efficiency of the system, and the application(s). Thus data model design and choice of data structure should involve considerable thought, and should not be taken arbitrarily. This is particularly true for an application-specific terrain model, if the DEM will need future modifications.

Efficient and flexible data structures are important to the development of computer mapping. Peucker & Chrisman (1975) state that most data models are characterised by structures which are convenient at the input stage rather than at the stages of use within a computer program (ie. little data manipulation within systems); separate and uncoordinated files for different types of geographic features, resulting in time consuming efforts to combine them; and a lack of information about neighbouring entities. These points may be abbreviated to flexibility, comparability, and topology, all of which may be considered a hindrance to GIS development (Peucker & Chrisman, 1975).

The performance versus representational fidelity tradeoff directly determines the storage, manipulative and retrieval characteristic of the data structure and physical file structure. Peuquet (1984) suggests examining these tradeoffs, utilising a specific set of usage-based criteria, so that the overall quality or suitability of a specific data model can be evaluated within a particular context. These general criteria are completeness, robustness, versatility, efficiency and ease of generation. Completeness may be thought of in terms of the proportion of all entities and relationships existing in reality, which are represented in the model. Robustness is the degree to which the data model can accommodate special circumstances or unusual geographical instances. Versatility is as the title implies, a measure of the model's adaptability to specific applications. Efficiency includes both compactness (storage efficiency) and speed of use (time efficiency). Ease of generation is the amount of effort needed to convert the required data into the form required by the data model.

In varying degree, each of these factors enters into consideration for any given application. The relative importance of each is a function of the particular type of data to be used and the overall operational requirements of the system. It is possible to measure quantitatively the performance of several of these criteria, eg. speed and space efficiency for a particular data model. However, it is not easy to provide quantitative measures for the completeness, robustness and versatility of a model.

Since the performance and efficiency of data structures may be considered in relation to data retrieval from main memory and external disk storage, this study shall only consider the former. Hence the results do not consider access to multiple DEMs.

2.2 Spatial Data

The term 'spatial data' applies to any data concerning phenomenon areally distributed in two, three, or n-dimensions. Geographic data, given the tendency of natural phenomena, occur in irregular, complex patterns and have traditionally been presented for analysis by means of

two-dimensional analogue models, known as maps. The map is a convenient method of spatial data storage for later visual retrieval and subsequent manual updating, measuring or other processing, but in order to do this, a new map must be drawn by hand, or the old one modified by hand. This process is both laborious and time-consuming. Other basic types of spatial data have evolved which are adapted to digital storage (Peucker, 1979; Peuquet 1984).

However, whilst features such as mountains, rivers or roads can be distinguished easily from a map, in digital form this is not the case. All entities are recorded in geographic coordinates such as latitude and longitude or in grid format. The definitions of the relationships between spatial entities, and the entities themselves, tend to be inexact and context dependent. Hence, topological relationships need to be derived from such file structures. For example, the relative location of a feature with respect to others can be simply determined from a map, but can become a time-consuming process in many digital terrain models. The combination of these properties make the modelling of geographic data uniquely difficult, since computer memory is one-dimensional in nature. The coordinates must therefore be structured so as to preserve the relationships, and yet be capable of being stored in linear fashion. For these reasons, the representation of neighbourhood relationships is a critical feature for efficient computer search (Peucker & Chrisman, 1975; Mark, 1978b; Little, 1978). Burton (1978) discusses techniques for the retrieval of geographic information on the basis of location. Topographical surfaces have implicit topological structures which are of value in many applications, but the potential advantages have neither been recognised or exploited (Mark, 1978b). The adjacency relations among points, lines and regions, that is topological structure, are an important element in the theory and implementation of terrain models for GIS (Little, 1978).

Peucker (1978, 1979), Pratt (1979), and Peuquet (1984), amongst many others, have attempted classification of terrain models by the nature of the data structures employed. From these reviews, it is possible to distinguish digital terrain models as follows :

- (i) **Point Structures** - Each data element is associated with a single location, such as information rich points, ie. peaks, pits, saddles, breaks, ridge and channel points.
- (ii) **Vector or Line Structures** - Similar to point structures since lines are defined by a series of points. The basic logical unit corresponds to a line on a map, such as a contour, river network, profile, etc. A series of X,Y coordinate point locations along the line are recorded as the components of a single data record.
- (iii) **Tessellation Structures** - The basic logical unit is a single cell or unit of space, most frequently based on a regular rectangular grid. Models include any infinitely repeatable pattern of a regular polygon or polyhedron.
- (iv) **Patch Structures** - The approximation of surfaces by a mathematical function is a logical step for mathematicians. Topographic terrain is too complex for any area of reasonable size to be represented by one function, so surfaces are usually partitioned into patches.

- (v) **Other Structures** - These include hybrid structures which possess characteristics of other types such as both vector and tessellation data. For example, a hybrid may consist of a regular grid with break lines and spot heights (Kostli & Sigle, 1986). Another different type of structure is discussed by Makarovic (1976), which is based on a regular grid DEM of variable sampling densities. A coarse grid is intensified with grids of higher densities in areas of rough terrain, using a progressive or composite sampling technique (Makarovic, 1973, 1975, 1977). However, the model may be converted to a uniform dense grid.

2.3 DEM Classification

The literature on DEMs (and surface representation) is immense, so classification is not straightforward. Methods overlap or conflict with one another, and there are differing approaches to categorisation. The task is not made easier by the fact that DEMs can exist on different levels of a classification. For example, a regular grid DEM could be used as the basis for constructing a polynomial patch model. For the purposes of this research, a two-part classification is attempted (see Figure 2.2).

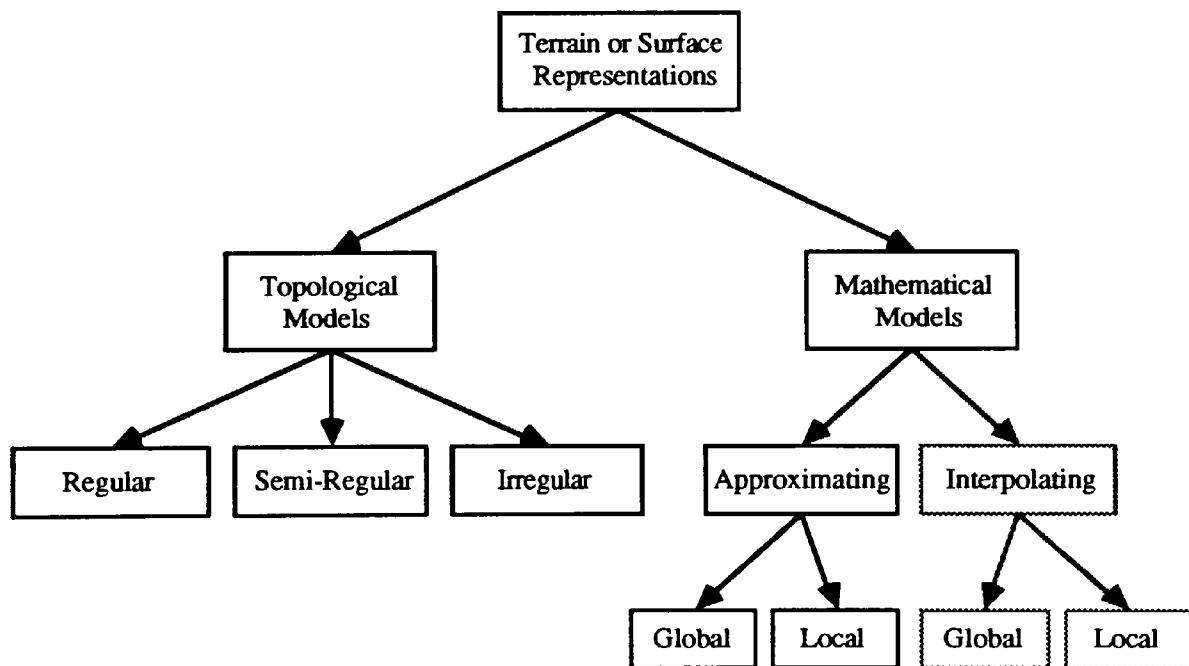


Figure 2.2 - Classification of Digital Elevation Models.

The first category outlines DEMs that only store point information, derived from the various sampling processes of the data acquisition techniques outlined in Chapter One. These are discussed with respect to their data structures and applicable interpolation techniques. The second classification approach attempts to identify mathematical methods of surface representation. These surface functions are derived from the terrain data supplied in the form of a point DEM. By not categorising mathematical methods on the spatial pattern of their source data, the classification becomes simpler. However, there are numerous mathematical

techniques for surface representation, so only the major methods and their basic principles will be discussed in any detail. There may well be some conflict when discussing how interpolation is achieved using the data structures of the point DEMs with the specific mathematical methods of surface representation, since the same techniques may sometimes be employed.

2.4 Topological Point DEMs

Point grids can be classified into regular, irregular and semi-regular. The term grid implies a network of values arranged in a mesh (most often regular) and calculated in such a fashion that the values at the nodes (where a given row and column intersect) are accurate samples from the surface. Regular grids have a low adaptability to the terrain variability. Since interpretation is not required, the sampling of regular grids is objective. Irregular grids are highly adaptable to terrain variability, but at the cost of intense interpretation. Sampling is thus subjective. Semi-regular grids have considerable adaptability to the terrain, and as interpretation is not required, sampling is objective.

2.4.1 Regular Grid DEMs

DEMs of this type can be generated from terrestrial or photogrammetric surveys, or as the result of the interpolation of irregular-spaced data, such as contours or other information rich lines. The sampling of the grids may be :

- (a) **Square/Rectangular** - These are the most widely used of all DEM data structures, since they lend themselves to many processes and operations, and to producing displays of all kinds. The programming for gridded data is natural, as is the application of matrix techniques. The topology is implicit and there is an additional economy in that (X,Y) coordinates can be calculated when needed from the (i,j) locations of Z value elements in the matrix, rather than kept in storage. The reason for the programming simplicity is an outcome of the natural spatial symmetry of the regular grid, and also of the implicit topological structure of the neighbourhood relations of each point in a grid. The grid structure allows adjacent grid cells, links or points to be found with few instructions, and locations in storage are instantly calculable from position. The obvious shortcoming of the grid is the redundancy of data required to store uniform areas, since the grid resolution must be set small enough to capture the variability required in rough areas.
- (b) **Triangular** - A characteristic unique to all triangular tessellations, regular or irregular, is that the triangles do not all have the same orientation. This makes procedures involving single cell comparison operations, which are simple to perform on the other types of tessellations, much more complex. Nevertheless, this same characteristic gives triangular

tessellations a uniqueness in representing terrain and other types of surface data, since height values are assigned to each vertex in the grid.

- (c) **Hexagonal** - The primary advantage of the regular hexagonal mesh is that all neighbouring cells of a given cell are equidistant from its centre-point. Radial symmetry makes this model advantageous for radial search and retrieval functions, unlike the square grid where diagonal neighbours are not the same distance away as its cardinal neighbours. However, Samet (1984) points out that hexagons limit basic resolution because, unlike rectangles, hexagons cannot be further subdivided into smaller hexagons. Instead hexagons can be grouped into aggregates or rosettes. Van Roessel (1988) provides an algorithm to convert from the rectangular to the hexagonal representation.
- (d) **Rhomboidal** - This is a very similar representation to the regular square grid, with the same advantages and disadvantages. This structure can also be regularly tessellated.

In terms of processing efficiency, the algorithms devised for square grids, can easily be modified to work in the cases of hexagonal, triangular or rhomboidal meshes. However, these grids are less frequently used for the representation of spatial data, even though the source of data origin may be the same, ie. interpolated or sampled photogrammetrically. The original problem of data redundancy still exists in these models, since the rectangles, triangles and hexagons are regular, throughout the model. Ideally, to overcome this problem, the model needs to be irregularly tessellated, to allow more data values to be stored in one area than another, ie. adaptable to terrain variability. The critical difference between these tessellations on the plane is that only the square and rhomboidal grid can be recursively subdivided with areas of both the same shape and orientation. Triangles can be subdivided into other triangles, but the orientation problem remains. Hexagons cannot be subdivided into other hexagons.

In general the disadvantages of regular grids can be summarised as:

- (1) Data redundancy,
- (2) Inability to adapt to terrain variability,
- (3) Constraint of grid size versus accuracy,
- (4) Loss of accuracy for interpolated grids (especially from digitised map contours),
- (5) Inability to represent specific terrain features.

The regular grid DEM, despite being at the forefront of terrain data models for the last thirty years, is not necessarily the most efficient in terms of cost-effective storage, due to this data redundancy and inability to adapt to the terrain variability. However, data compression techniques exist that can remove the redundancy within data such as regular grids, but at a cost of increased time for data access. One such method is Huffman coding (Huffman, 1952; Williams, 1986), which is a variable length coding scheme that assigns short codes to

frequently occurring values and longer codes to sparse data on the basis of some statistical criterion relating elevations, thus adapting itself to terrain variability with no loss of accuracy. This could be achieved by using an algorithm to predict grid elevations, for which the differences from the actual elevations could be Huffman-encoded. Other grid techniques can be used to approximate dense grids, such as the use of a fixed difference altitude between points, termed a 'microgrid' (Boehm, 1967).

2.4.2 Semi-Regular Point DEMs

Semi-regular point DEMs attempt to retain some of the topological attributes of grid-based techniques, such as neighbouring entities and/or a similar coordinate system, but aim to reduce the inherent data redundancy of such schemes.

- (a) **Drop Lines** - These models are derived either from digitised contours by parallel scanning at regular or irregular intervals, or alternatively from photogrammetric methods (Yoeli, 1975). The point distribution of these is very advantageous. They are more economical than regular grids, since the points are selected at characteristic points of the profile only (eg. gradient change). This method causes the extreme points of the profiles to be contained in the data. There are two types of drop line models - data points selected at salient features or at constant vertical intervals. Interpolation algorithms can be implemented by considering there to be a net of triangles in every strip of terrain between two adjacent profiles, or alternatively, by using simple distance-weighted interpolation techniques. The disadvantage of this system is that only the X-coordinates are implicit in terms of data storage, so the resulting model needs to store the Y-coordinates, together with the X-increments and heights. Nevertheless, the storage savings in omitting redundant points compensates for this.
- (b) **Heterogeneous Square Grids** - One way of overcoming the problem of data redundancy is to use a grid of variable density, where the density of the acquired grid points is adapted to the terrain variability. This solution has been derived from photogrammetry in the form of progressive and composite sampling (Makarovic, 1973, 1975, 1977). Instead of all the points in a dense grid being measured, the density of the sampling is varied in different regions of the grid, being matched to the local roughness of the terrain. Starting with a low resolution grid, a progressive increase in the density of the sampling takes place on the basis of an analysis of terrain relief and slope using an on-line computer attached to the photogrammetric instrument. Thus the grid is increased in density by halving the size of the grid-cell in certain limited areas, based on the terrain analysis. Subsequently, an increased density of points may be prescribed for still smaller areas. Normally, three such runs are sufficient to acquire the data to define a satisfactory model.

In this way, the progressive sampling technique attempts to optimise the relationship between specified accuracy, sampling density and terrain characteristics. A terrain model based on these techniques is described by Makarovic (1976).

A similar approach is used in grid-based terrain modelling packages such as HIFI (Height Interpolation by Finite Elements); SCOP (Stuttgart COntour Program); and GTM (Graz Terrain Model). The SCOP DEM is stored essentially as Z-values of a rectangular grid, where the grid lines run parallel with the coordinate axes (Kostli and Sigle, 1986). Form lines which describe other characteristic shapes in the terrain can be intermeshed with the grid, thus enabling a strict consideration of the most important terrain form elements for DEM applications. Kostli & Wild (1984) give an example of a heterogeneous square grid, which reduces points by up to 50% for a test model.

- (c) **Surface Patch Quadtree** - An extension of variable density grids is the surface patch quadtree. A quadtree is a hierarchical data structure that is able to focus on subsets of the data, resulting in an efficient representation, particularly for set operations. Many of these operations could be performed equally as efficiently with other data structures, but quadtrees are attractive because of their conceptual clarity and ease of implementation. They are based on the principle of recursive decomposition of a region into four equal sized quadrants, and can be differentiated by the type of data that they are used to represent. Peuquet (1984) discusses the advantages of quadtrees for geographic phenomena. The area is subdivided into quadrants and sub-quadrants etc., until the region has been completely defined. This process is represented by a tree of degree four, each non-leaf (grey) node having four sons. Leaf nodes that represent the region are termed black, whilst non-region representing nodes (empty) are termed white. The root node corresponds to the entire region and each son of a node represents a quadrant labelled in order NW, NE, SW, SE. A surface patch quadtree will represent the whole of the region (or surface), such that there are no white nodes. Instead of black nodes representing a sub-region whose attribute is implicit, a mathematical function defining that particular patch is stored, together with a key, indexing the coordinates and size of the node.

Martin (1982), Chen & Tobler (1986) and Leifer & Mark (1987) have developed surface patch quadtrees based on various mathematical functions. Tolerance levels for the maximum allowable error or accuracy are set before the quadtrees are created. A linear quadtree of the surface can then be constructed, starting with a root node covering the whole surface and fitting a function to the data points (Gargantini, 1982; Abel & Smith, 1983). The heights at the original points are then interpolated from this function, and if the tolerance level is not attained, the node is sub-divided and placed in a queue. The sub-division continues until a node's function satisfies all the points in its patch, for all

nodes. The functions used to represent this patch include orthogonal polynomials (Leifer and Mark); and average, upper, lower, ruled and quadric surfaces (Chen and Tobler).

- (d) **Heterogeneous Triangular Grids** - Just as square patches can be represented hierarchically by repeated subdivision (using quadtrees), it is possible to tessellate triangles of variable precision by the same principle (Sotomayor, 1978). Initially, the region is split into four tilted triangles. If the surface approximation by a given triangle does not meet the desired precision, it becomes subdivided into four sub-triangles, which in turn will be subdivided if they do not satisfy the error criterion. The drawbacks of time inefficiency and database access are overcome by tessellating the surface with triangles in an ordered and hierarchical way, whilst structuring the data files as a quaternary tree. The points of the surface are information rich points, and therefore random. During a subdivision, the nearest surface-specific points to the mid-points of the triangle's sides are selected, or alternatively, the mid-point becomes the new vertex. However, the vertices of the final model may not all necessarily correspond with the original data set, since the triangulation is constrained in some form to a regular tessellation.

Barrera & Vazquez (1984) also describe a heterogeneous, hierarchical triangulation method for representing terrain relief. Each triangle is referenced by a unique variable length key which relates to the coordinates of each vertex. The generation of triangles is accomplished by partitioning on edges. However, despite its reported suitability for applications such as line-of-sight calculations, its computational efficiency may not be as good as traditional irregular triangulation data structures.

- (e) **Hierarchical Tessellations** - These have developed from the desire to model continental or planetary data within a GIS. An extension of hierarchical triangulation is the Geodesic Elevation Model (GEM) (Dutton, 1984). GEM is designed to digitally archive and access measurements of points given in latitude, longitude and elevation by embedding them in a regular polyhedral data structure. Whilst encompassing features of the regular grid and triangulated irregular network (TIN) (see section 2.4.3), the model is designed to be planetary in scope. Difference-encoding of elevations provides a compact, self-calibrating and scale-sensitive representation of topographic relief (Dutton, 1991). A scheme derived from GEM is the Quaternary Triangular Mesh (QTM), which is a region quadtree composed of triangles. It represents a planet as an octahedron comprised of eight quaternary triangular grids, and can encode locational data both as hierarchies and sequences (Dutton, 1989, 1990, 1991). A similar tessellated scheme for the globe is described by Goodchild & Yang (1990). Such data structures are at an early stage of development but have the potential to provide the framework for handling the vast amounts of DEM data that are becoming available.

All of the methods discussed attempt to overcome the data redundancy problem of regular grids, by being more adaptive to the characteristics of the terrain. It can be seen that in doing so, more complex data structures are required, and interpolation algorithms will be more complex than for a regular grid, especially for searching operations, since some of the implicit topology of the regular grid is lost. To compensate for these problems, and for a method to be considered practical, it must be able to finely adapt itself to the terrain to produce large savings in storage (or rather store a minimum of points). The methods that appear the most promising in this respect, and which are adaptable to terrain variability whilst maintaining some of the regular grid's topological features are the surface patch quadtree and the heterogeneous or variable density grid.

2.4.3 Irregular Point DEMs

Methods which only consider the critical points of the terrain, such as the information-rich points and lines, will be the most economical in terms of storage. However, storing points arbitrarily will further exaggerate some of the problems of semi-regular DEMs, namely structuring the data and increased complexity of the interpolation applications due to the lack of topological relationships. Since there are no direct spatial relationships between points, it is generally accepted to derive the relationships after point selection, most commonly by considering the data to be at the nodes of a network, or alternatively using a local spatial referencing structure, such as a grid or quadtree (Burton, 1978; Samet, 1984). However, data storage increases dramatically when the derived spatial relationships are included in the model.

Irregularly sampled data overcomes many of the disadvantages concerned with regular data. Since there are no limitations imposed on the spatial distribution and relationship of points, all the important features of the terrain can be incorporated. The problem of data redundancy does not arise, since no superfluous data would be included. The data could be extracted from a dense regular grid DEM, thus overcoming the problems of data acquisition, namely time and cost. Many algorithms have been developed for this purpose, especially for hydrographic identification of watersheds and water-basins. A review of this literature and results is presented in Chapter Eight.

Irregular DEMs can be distinguished as:

- (a) **Point Quadrees** - Point data can be represented in a variety of ways, depending on the applications to be performed. The point quadtree is a multi-dimensional generalisation of a binary search tree. In two dimensions, each data point is a node in a tree having four sons, which are roots of subtrees corresponding to quadrants. The process of inserting into

point quadrees is analogous to that used for binary search trees. In essence, the desired record is searched for on the basis of its X and Y coordinates. At each node of the tree, a four-way comparison operation is performed and the appropriate subtree is chosen for the next test. Reaching the bottom of the tree without finding the record means that it should be inserted at this position. Point quadrees are especially attractive in applications that involve search. The efficiency of the point quadtree lies in its role as a pruning device on the amount of search that is required. Thus many records will not need to be examined. They are an effective structure for handling point data, but it remains to be seen how well they can handle geographic data consisting of thousands of points. Samet (1984) has given an extensive review of the literature on quadrees, but with no detailed performance evaluation.

- (b) **Contour DEMs** - Contour representations of terrain (Boehm, 1967; Merrill, 1973) are also known as polygonal DEMs, where points are stored as open or closed strings. The economy of the contour method is related to adaptability. Where a surface is smooth, few contours are needed to represent it, and these may be described by a relatively small number of points, if a line generalisation algorithm has been applied. Where the surface is rough, more contours are necessary, and since these will be more convoluted in these regions, more points are required to represent the lines. The encoding of surfaces by contours minimises the storage capacity, whereas a regular grid of surface points minimises the computing time necessary for several types of manipulations (Boehm, 1967). A disadvantage of the contour DEM is that breaks are not usually incorporated. These are structure lines, gradient changes, and peaks, pits and passes, which do not usually occur precisely at the height of a contour.

Storing contours and maintaining topological relations has been recognised as a problem, since no single method clearly distinguishes itself from others, whilst methods may or may not be efficient for computer search applications (Merrill, 1973). For most applications, the formulation of the logics of corresponding computer programs can be rather tricky. For a set of contours it is relatively easy to create a directory which indicates a sequence of contours in a type of tree, in which the surrounding contour is the base and the other contours are the branches (Boehm, 1967). However, the topological relationship between contours and their points make some operations difficult. Point interpolation involves accessing at least two adjacent contours, searching through the data and calculating the distances to the point. This can be very time consuming if the contour lines are represented by many points. Most algorithms use only the closest n points, by considering the data as a set of randomly distributed point observations, but because point spacing along contours will be less than the distance between contours, interpolation will tend to produce flat areas near the contours.

Clarke et al. (1982) discuss four groups of algorithms for interpolation based on the methods employed to locate contour data points. Yoeli (1986) and Cole et al (1990) describe methodologies for the interpolation of points from contour strings to a regular grid DEM. Another approach is to use the contour points to form a spatial net of triangles. Triangulation of contour points has been implemented by McCullagh (1983), Christensen (1987) and Scarlatos (1989), amongst others. These procedures are more effective if elevations are also sampled at critical surface points. However, it has long been realised that such sampling represents a generalisation, and fails to capture all of the information present in the contour map. If contour lines are first digitised at a sufficient density to capture all of their irregularities (and if spot heights are also digitised), then all of the topographic information present on the contour map will be in machine-readable form.

An alternative method for computing, storing and retrieving contour information is described by Rom & Bergman (1986). Contour maps are pre-processed in order to represent contours in a piecewise parametric form, stored in an organised manner for efficient access.

- (c) **Topological Planar Surfaces** - Geographic entities on planar surfaces are points, lines and polygons, such as contours. The simplest system is that of encoding each individual entity, but there are limitations with spatial relationships. Peucker & Chrisman (1975) review some of the most popular data structures with specific regard to maintaining topological relations. Systems have been developed based on a common location dictionary which contain the coordinates of every point on the map, such as the 'spaghetti model' (Peuquet, 1984). This reduces the search time compared with the entity by entity approach, but the neighbourhood relationships are not well defined. By adding the topological neighbourhood function of each element to a data structure, improvements in flexibility and scope of applications can be realised. This has been termed the 'topological model' (Peuquet, 1984).

The DIME (Dual Independent Map Encoding) model represents neighbourhood relations in the form of tables. Line segments are encoded, with the names of the polygons to the left and right of each, together with the labels of the two end-nodes. Peucker and Chrisman (1975) discuss the POLYVRT data structure, which is similar to DIME, whose basic object is a 'chain' made up of many points, as opposed to the two point lines of the 'topological model' and DIME, thus allowing lines to be represented by a single chain. In addition to the indication of the relative location of the chain with respect to its neighbouring polygons, POLYVRT information is stored in separate lists assembling the bounding chains for every polygon. Bouillé (1978) presents a topological model HBDS (Hypergraph-Based Data Structure), which is based upon set theory and the hypergraph concept. Burrough

(1986) also gives descriptions of some polygon methods and some of the operations involved with them. The arrangement of the data within such systems and the amount of explicit storage varies, but the principle relies on the availability of all adjacency information (Little, 1978).

- (d) **Quadrilateral DEMs** - Shmutter and Doytsher (1978) believe that quadrilateral DEMs eliminate most of the disadvantages which are inherent in a regular grid DEM. Grid lines are determined in accordance with existing topographic data. Net meshes are of varying shape and size, their area becoming small in regions with dense given information and expand in size in regions where the data is sparse. The procedure for determining the grid is founded upon the concept of information density, the aim being to produce a pattern which ensures a uniform flux of information density over all grid cells. This method attempts to shape the grid lines of the quadrilateral to the surface points, although not necessarily joining the points. There are other modelling systems based on quadrilaterals, but these are more commonly used in Computer Aided Design (CAD). Such methods require local blended surface patches to be fitted to the data by some mathematical constraint (Barnhill, 1977; Tipper, 1979).
- (e) **Triangulated Irregular Networks (TIN)** - The TIN is the most common alternative to the regular grid DEM. Its main advantage is that every measured data point is honoured directly within the model, since they form the vertices of the triangles. Furthermore, the use of triangles offers an easy way of incorporating break-lines, faults, ridges and channels. The TIN is very similar to a topological system, since they divide the area of coverage into irregularly shaped regions and rely on the explicit adjacencies of these regions to organise the spatial data (Little, 1978). Various criteria for a good triangulation have been defined, but many take four points forming the vertices of a quadrilateral and decide which of the two possible internal diagonals is preferable. Criteria include maximising the minimum height, maximising the minimum angle, and minimising the diagonal length (Gold, 1979). To ensure these conditions were met, the triangulation is constructed iteratively, by starting with an arbitrary arrangement of triangles and adjusting the mesh until an optimum configuration was obtained. However, this may result in exorbitant run times.

The problem has been overcome with the development of algorithms that produce optimal networks in one pass, such as the Delaunay triangulation, which is uniquely defined for a given set of points. In addition, the triangles are as nearly equiangular as possible, and the longest sides are as short as possible. McCullagh and Ross (1980) describe an algorithm to achieve this, and show that it is unique, irrespective of starting position. Watson (1982) describes an iterative algorithm to create a Delaunay triangulation for a

contouring application. Mirante & Weingarten (1982) describe an alternative method to the Delaunay triangulation, known as the 'radial sweep algorithm'. McKenna (1987) uses an inward spiral triangulation, which also generates an optimal network in one iteration.

Fowler and Little (1979) use an error-tolerant TIN, derived from a regular grid, showing that a substantial saving in the number of points in the model can be achieved, compared to the regular grid. Peucker et al (1978) state that the point ratio between the TIN and grid is approximately 1:7 points in the worst case when the terrain is of very high relief, and 1:200 points in the best case, when the terrain is relatively flat. Herein lies the advantage of the TIN over the regular grid. These results though, do not take into account the overheads of storing the actual triangular network.

McCullagh (1983) uses the PANACEA system to combine both the advantages of the TIN and the regular grid, when both speed in execution and convenience of use are required. The system can calculate, edit and store a triangular network from contours and other critical points to produce a regular grid DEM, but at the resolution specified by the user. Christensen (1987) also triangulates contour points, but highlights the failure of the Delaunay triangulation to connect points in the shape of contours. A case is that of a triangle edge crossing a contour segment, leading to large errors. This can be avoided by performing Delaunay triangulations in between adjacent contours, thus creating many triangulations independently of each other. Another problem encountered is one in which a triangle has its three vertices on the same contour, causing breaks in the surface and a poor configuration of triangles. Christensen proposes using a medial axis transformation, since there are always points on the axis that can be connected to the contour points, and furthermore because it is executed between contours, the triangulation does not cross them. Scarlatos (1989) has also recognised these problems, but believes algorithms suffer because they require extensive human interaction; do not extend well to complex terrain; or in Christensen's solution, double the number of data points as a side effect. Her algorithm, sorts the data and then decomposes it into a series of trapezoids, which are then split by new edges linking points on the trapezoids. Finally, all of the resulting edges, new and old, are used to define the triangular mesh.

Most applications of the TIN involve one or more of three basic processes:

- (1) Sequential element by element processes.
- (2) Searching for the closest node to a point, or locating a point within a triangle.
- (3) Intersection of the terrain surface with various other surfaces.

One of the most popular applications of TINs is for the contouring of irregularly sampled

points, since it avoids the disadvantages of generating the more conventional regular grid DEM (Gold, 1987; McCullagh, 1988). Peucker et al. (1978) illustrate the wide range of TIN applications, including automated hill shading, slope mapping, contouring, profiling, line of sight maps, and integrating TIN with coverage data - all applications which may be incorporated into a GIS. Gold (1978) and De Floriani (1987) present overviews of triangulation, with specific regard to data structures, implementation and applications.

- (f) **Other Irregular Methods** - Other ways of representing irregularly spaced data assume no spatial relationships exist with neighbouring points, but are stored in a locally referenced data structure such as a grid or quadtree (Burton, 1978; Samet, 1984, 1990a, 1990b). This type of DEM can be used for applications such as interpolation, where a reference of local points is maintained, and can be easily accessed. Alternatively, it can be used as the basis for the formation of other DEMs, such as the TIN, since a list of local reference points will reduce database access time. Surface-specific or critical points form the basis for many irregular DEMs, since they are more information-rich than other points.

One method of data storage is termed the 'box structure' described by Knuth (1973) and McCullagh & Ross (1980), which divides the surface into boxes or patches. The file of points (stored as X, Y, Z coordinates) is sorted, firstly into descending Y-order, and then by X-order within the limits of each overlaid grid cell. The data file is thus a succession of points in an ordered listing, such that each row of boxes is in Y-order, but the data inside each box is sorted by X-order. An index can indicate the position in the file of the first point within each grid cell. The larger the data set, then the more efficient the box-structure will be in terms of access time, compared to an unstructured set of arbitrary points. The question which arises most commonly regards the optimum average number of points per patch for greatest efficiency. A similar 'sortedcell matrix' method is described by Hodgson (1989) for rapid grid interpolation.

2.5 Mathematical Surface Models

The determination of a surface defined by regularly or irregularly spaced data can be stated as :

"Given the points (x_i, y_i, z_i) , $i=1, 2, \dots, n$, over some domain, a function $z=f(x,y)$ is desired which reproduces the given points, and produces a reasonable estimate of the surface (z) to all other points (x,y) in the given domain" (Schumaker, 1976).

There are two general approaches to surface modelling via irregularly spaced data, namely approximation and interpolation (Schumaker, 1976; Barnhill, 1977). Approximation is applied when the data are considered to be 'noisy', or have some measurement errors, and is regarded as data smoothing. Least squares based algorithms are common in these techniques.

Interpolation is generally used when the data represent 'exact' values, and/or there is a need to ensure that the model reproduces the initial data. Furthermore, approximation or interpolation methods can be classified as being global or local. Global methods are those in which all of the data participate in determining the modelling function, whereas local methods are those in which the modelling function is determined by data 'nearby' where the function will be evaluated. Numerous interpolation and approximation methods exist for both global and local methods. Regularly spaced data is more usually represented by approximation methods rather than interpolation techniques, since as well as smoothing the data, they will produce a good deal of data compression. A feature of some mathematical methods is that they can be used as both interpolation and approximating techniques, such as polynomials and splines. This review will concentrate particularly on approximation techniques.

Tipper (1979) has identified both inherent advantages and disadvantages with global and local strategies :

- (a) A global representation needs only one set of functions for the whole surface, giving savings in computational time and storage. However, they require extremely complex functions, even if the tolerance allowed in the model is high. The resulting surface will often be unreliable at its extremes. Furthermore, if it is at all complex, it will often behave erratically as an interpolant.
- (b) Local representation is computationally less efficient in terms of storage required, as one set of functions must be stored for each patch. Each set of local functions can, however, be relatively simple and yet still have a good representation. The simplicity of the functions and the fact that they are locally based imply that a local representation is often highly efficient for large data sets.
- (c) A global representation is continuous to the same degree everywhere. To achieve this result using local patches, the form of the functions is severely restricted and the number of patches may have to be increased.
- (d) It is often impracticable to use a global representation for multiple-valued or concave surfaces. There is no such restriction for the piecewise approach.

Wolf (1991) attempts a characterisation of mapping functions which may be regarded as abstract models of topographic surfaces. This paper raises a number of important points, including the fact that continuously differentiable functions do not necessarily represent realisable topographic surfaces due to unlikely peculiarities.

2.5.1 Global Methods

The use of global fitting methods presupposes the existence of a simple surface throughout the data area, or the desire to simplify the representation of a complex surface within that area. These methods involve the fitting of a single three-dimensional surface defined by a high-ordered equation, through all of the measured randomly located terrain points. Once this

global surface has been defined and its coefficients calculated, points may be interpolated by a simple substitution of its coordinates.

(a) Trend Surface Analysis

This covers a wide range of related numerical methods, whose common purpose is the objective description of broad-scale spatial trends in mapped data. It is a method of separating data into two components - that of a regional nature and local fluctuations. This is accomplished by fitting a trend function to a set of data values. Most trend surfaces consist of a plane or gently curving surface defined by integer power series polynomials, usually by least-squares. This results in the trend surface passing through, above, or below each actual data point. The difference between the computed value of the trend surface at a point and the value of the actual point is termed the residual. In satisfying the least squares criterion, the sum of the squared residuals is minimised.

Any scalar field can be represented by the equation $Z_i = f(X_i, Y_i)$ relating surface height (Z_i) to position (X_i, Y_i) at each and every data point. Trend surfaces specify the precise mathematical form for this function and fit it to the data by least squares regression. It is unlikely that any simple function will exactly fit the data, because it is exceedingly unlikely that only one trend-producing process will be in operation, and there are likely to be inaccuracies in the measured data. Whitten (1975), has attempted a classification of the most common trend surface analysis techniques :

- (a) Orthogonal algebraic polynomials for regular data (Oldham & Sutherland, 1955; Grant, 1957; Krumbein & Graybill, 1965) and irregular data (Whitten, 1970).
- (b) Non-orthogonal algebraic polynomials (Harbaugh & Merriam, 1968; Cliff et al., 1975; Unwin, 1981; Lam, 1983; Davis, 1986).
- (c) Double Fourier series or trigonometric polynomials (Harbaugh & Merriam, 1968).

Despite the computational and mathematical difficulties involved with these methods, the main problems in the practical use of trend-surface analyses are conceptual, for example, fundamental questions involve how 'good' a particular trend surface is and the degree of similarity for two surfaces. This relates to the completeness, robustness and versatility measures of a data model discussed by Peuquet (1984). However, in trend surface analysis, this comparison is usually accomplished statistically, by expressing the proportion of the total corrected sum of squares of the mapped data (Harbaugh & Merriam, 1968; McCullagh, 1973; Davis, 1986).

- (i) **Orthogonal Algebraic Polynomials** - This method consists of fitting a polynomial to the data up to the point where the trend is adequately specified. The success of this approach

depends upon the validity of the assumption that whatever trend exists can be adequately described by a polynomial of low order (Grant, 1957). For gridded data, the arithmetic involved in solving for the coefficients is relatively easy. The major advantage of the method is that coefficients can be added to the equation, since each one acts independently of each other. Surfaces can be modelled to a specific goodness of fit, which other mathematical models have limited control over. However, the form of the equation has no X and Y coefficients so points are interpolated as a sum of the coefficients multiplied by the vectors corresponding to the orthogonal polynomials, which can result in a lengthy approximation process, especially for larger equations.

In the late sixties and early seventies, the benefits of using orthogonal polynomials for trend surface analysis of irregularly spaced data were recognised (Whitten, 1970). This technique involves simple arithmetic operations that are executed extremely fast, since no matrix inversion is necessary. It is impracticable to construct a general set of tables for irregularly spaced data, because the data depend on both the number of points and the varying interval between each of the points. The method consists of defining vectors made up of individual terms for every point in the data set, so that a set of polynomials can then be defined. It is required that all of these polynomials be orthogonal, so it is necessary to calculate the values of the coefficients that establish this condition. Once accomplished, the polynomials can be defined and an equation for the surface formulated.

- (ii) **Non-Orthogonal Algebraic Polynomials** - A major problem in any trend surface analysis is the decision upon a particular function for the trend part of the equation. The most popular method takes the form of any of the terms of the general polynomial equation:

$$Z_i = a_{00} + a_{10}x + a_{01}y + a_{20}x^2 + a_{02}y^2 + a_{11}xy + a_{30}x^3 + a_{03}y^3 + a_{21}x^2y + a_{12}xy^2 + \dots + a_{jk}x^jy^k,$$

where Z_i is the height of an individual point i ; x, y are the coordinates of each point and $a_{00}, a_{10}, a_{01}, \dots, a_{jk}$ are the coefficients. Each term will reflect a particular shape, so that the coefficients will determine the degree of influence for each term. Each coefficient makes a specific spatial contribution to the form of the final surface. A least-squares regression ensures that the coefficients of this equation have values which collectively make the sum of squares of the residuals all small as possible. Surfaces of increasing degree can be created, which will produce significantly better percentage fits to the original terrain. Usually up to 10th degree surfaces are used, but the unpredictable nature of the oscillations produced by such high-order polynomials may cause poorly interpolated values (McCullagh, 1973). A major disadvantage of the method, is that the coefficients are not independent of each other, such that the identification or separation of a trend is not permitted.

An extension of Gaussian least-squares theory has led to the development of an efficient algorithm for computing the coefficients of polynomial surfaces. This is based on the powerful, dimensionally invariant concept of the high speed matrix generator (HSMG) which computes and stores only the minimum number of non-repeating terms needed to form the coefficient matrix (Balch and Thompson, 1989). This method is an extension of the one-dimensional polynomial HSMG (Balch and Thompson, 1988).

(iii) Double Fourier Series - Fourier Series can be used to describe one or two dimensional variation by modelling the observed variation by a linear combination of sine and cosine waves. The terminology associated with Fourier series is derived largely from electrical engineering and time-series analysis (Harbaugh & Merriam, 1968; Davis, 1986). A complex oscillating or repeating pattern, such as an electrical signal can be considered to be the sum of a large number of sinusoidal wave forms. Basically, Fourier series provide a means of separating a surface into a number of simple harmonics. The wavelength (distance from crest to crest) and amplitude (half the height from trough to crest) are properties of the individual harmonics. Although these are independent of each other, they may be added together to produce complicated surfaces (as with orthogonal polynomial surfaces).

The amplitudes and phase angles of these simple wave forms can be determined by fitting a series of harmonics of sine and cosine waves to the data. In an analogous manner, a complex surface can be considered to be the sum of two interacting sets of two-dimensional sinusoidal wave forms, each containing many harmonics of differing amplitudes and phase angles. Just as any profile or surface may be approximated by a polynomial of sufficiently high degree, so may any profile or surface be approximated by a Fourier series with a sufficiently large number of terms (Davis, 1986).

The use of double Fourier series is, in part, similar to that of polynomial surface fitting, since they both aim to satisfy a least squares criterion, permitting major trends to be identified, and residuals to be calculated. There are computational advantages in fitting Fourier series to a regular grid, namely the reduction in computation time, and the ability to fit series with a large number of terms with relative ease. The coefficients can be obtained through a numerical integration method in which the volumes of prisms defined by the grid cells and the heights of the observed surface are summed.

A principle weakness of using double Fourier series with regularly spaced data is that the fundamental wavelength in a particular direction is governed solely by the dimension of the sampling grid. Also the Fourier surface fails to accord with the actual surface at places along the edge of the map. Double Fourier series can be readily adapted to irregularly spaced data in a manner similar to regular polynomial surfaces.

The versatility of Fourier series for approximating continuously variable distributions and the probable inherent harmonic nature of many geological and geophysical phenomena would suggest that for many purposes, double Fourier series trend surfaces might be superior to polynomial surfaces (Whitten, 1975). Many natural features including some land forms, are more realistically represented by Fourier series, reflecting periodicities inherent in the actual features. In their capability of representing complexities however, the higher degree polynomials have certain advantages.

(b) Multi-Quadric Analysis

The hypothesis for multi-quadric analysis is that any smooth mathematical surface, and also any smooth irregular surface (mathematically undefined), may be approximated to any desired degree of exactness by the summation of regular (mathematically defined) surfaces, particularly quadric forms (Hardy, 1971, 1975). Functions are fitted around individual data points, thus it is initially a local fitting procedure, yet the evaluation of a height at a given point involves taking into account all the local functions, thus it is also a form of global fitting. The cost of evaluation of any height is therefore related to the total number of points defining the surface, which thus makes it unsuitable for extensive terrain or grid data. In general it is a slow technique, since the order of the matrix to be inverted increases linearly with the number of points.

(c) Kriging

Kriging is a distinctive form of interpolation, originally devised as a method of moving averages to avoid systematic overestimation of reserves in the field of mining (Delfiner and Delhomme, 1975). The statistical surface is regarded as a regionalised variable that has a certain degree of continuity. However, no matter how short the distance is between two points, their values are statistically independent of each other. The zone of influence around a point may not have the same extent in all directions. Yet there has to be spatial autocorrelation, that is, a dependence between sample values which decreases with their distance apart. These characteristics of regionalised variables are quantified by the sampling variances and covariances, that is, the autocovariance matrix, from which the Kriging estimates of unknown points are determined (Lam, 1983). Because different assumptions about the regionalised variables may be involved, two systems of Kriging procedures, simple Kriging and Universal Kriging have been distinguished.

Generally, simple Kriging has more restrictive assumptions, but fewer computational problems, whereas the assumptions of Universal Kriging are more general, but difficulty of calculation is greater. The Universal Kriging method is not reliable unless a very large number of points are available, but the improved accuracy will not always justify the computational effort. In situations where a statistical interpretation of the data is valid,

it has been determined that Kriging easily and systematically outperforms other interpolation methods. A more extensive review of Kriging is given by Burrough (1986).

(d) Spline Surfaces

Splines can be used as both approximating and interpolating functions for both regular and irregular data. For interpolating splines, knots are located at all of the data points. Splines offer many advantages, since they are simple, analytic and piecewise, hence involving relatively few points. Splines of low degree, such as bicubic splines, are sufficient to interpolate or approximate surfaces quite accurately (Lam, 1983).

Hayes & Halliday (1974) present a method for fitting bivariate cubic splines by least squares to arbitrary data. Bicubic splines are defined over a rectangle in the (X,Y) plane, with the sides being parallel to the X and Y axes. The rectangle is divided into panels, again by lines parallel to the axes. Over each panel the spline is a bicubic polynomial. Each of these polynomials joins the polynomials in adjacent panels with continuity up to the second derivative. The constant X-values of the dividing lines parallel to the Y-axis form the set of interior knots for the variable X, corresponding precisely to the set of interior knots of a cubic spline. Similarly, the constant Y-values of dividing lines parallel to the X-axis form the set of interior knots for the variable Y. The bicubic spline fit has the property that the sum of squares of its weighted residuals is as small as possible for the given knot sets. The method is adaptable to terrain (since knots can be prescribed until a satisfactory fit is obtained), but can be considered time consuming in setting up the model. Another advantage of the method, is that the weighted least-squares approach can enable specific points to be given a better fit than others, eg. for peaks and pits, provided that the important points can be identified.

2.5.2 Local Methods

(a) Moving Surface Methods

These methods require, for each interpolated point, the computation of a surface, whose weighted sum of squares of distances to the reference points is a minimum. This surface will change its orientation, and possibly its shape, from one interpolated point to an adjacent one. For this reason, it has been called a moving or roving surface (Rhind, 1975; Schut, 1976). The height of an interpolated point is found as the height of the instantaneous position of the surface at that point. The differences between these methods consist in the type of surface and the weight function that are used. The surface will be either a level plane, a tilting plane, or a second-degree surface. It is defined algebraically by one or more terms of the equation $Z = a_0 + a_1x + a_2y + b_1x^2 + b_2xy + b_3y^2$. For the

interpolation of a point, the coordinates and height of each of the surrounding reference points are substituted into this equation (each reference height is given a weight that is a monotonic decreasing function of the distance to the reference point), and the parameters are found by least squares. Taking the origin of the coordinate system as the interpolated point, its height is a_0 , and only this parameter is calculated.

The crudest method uses a level plane and simply equates this height with the height of one of the nearest reference points. This results in a discontinuous representation of the terrain by a set of level surfaces. A less crude representation is obtained by making the height of the level plane a weighted mean of the heights of the selected surrounding reference points. Better representations are obtained by using a tilted plane or a second degree surface. The latter can give large errors in cases of few reference points. The use of a third degree polynomial increases that danger and does not appreciably improve the results (Schut, 1976). The choice between level plane, tilted plane and second degree surface could be programmed to depend on the number of surrounding reference points.

If a moving surface is to be used for all interpolations, it must produce a surface that is continuous in areas where the terrain has this property. That can be achieved only by using the same formula for all interpolations. To keep the computation time within reasonable bounds, only reference points within a specified maximum from an interpolated point will be used. It is then necessary also to use the same maximum distance for all interpolations; not to restrict the number of points within this distance; and to use a weight function that approaches asymptotically to zero, or nearly so, when the distance to a reference point approaches the specified maximum distance. A number of different weighting functions can be used (Schut, 1976).

(b) Summation of Surfaces

These methods construct around each reference point a fixed surface that has a vertical axis of symmetry. In a specific method, these surfaces differ only in vertical scale. Interpolation is performed by summing the heights of all surfaces. The methods of this group employ an algebraic formulation that is similar to linear least-squares interpolation. Smoothing of the data (filtering) is also possible. A distance function, called the correlation or covariance function is either defined or computed. From this function follow the elements of a two-dimensional array, whose elements represent distance functions between reference points. The interpolated points can then be calculated. This interpolation can be expected to give better results than other methods, if the data have indeed the character of a stationary random function, and if the proper correlation function is used. To qualify as a stationary random function, one requirement is

that the systematic trend in the data be eliminated by reducing the heights to a suitable reference surface (Schut, 1976).

(c) Simultaneous Patchwise Polynomials

The region of interest is divided into square or rectangular elements by means of a coarse regular grid, such that the terrain surface is represented in each element by a low-degree polynomial. This is accomplished in such a way that the total surface is continuous and possibly smooth (ie. at their common boundaries, the local polynomial surfaces agree in height and possibly tilt). The common characteristic of these methods is the simultaneous computation of all the local surfaces (Jancaitis & Junkins, 1973; Pfaltz, 1975; Jancaitis, 1978). The hypothesis for these methods is that the weighted sum of squares of the residuals, plus a weighted sum that expresses the flatness of the total surface, must be a minimum. This condition leads to the formation and solution of normal equations by the method of least squares. The second of the two sums is introduced to ensure non-singularity of the matrix of the normal equations, and to flatten out the surfaces in poorly controlled areas. Each local polynomial may use up to 16 terms of the full bicubic polynomial, depending upon the degree of continuity and smoothness constraints of the total surface. The formation of this polynomial is described in Chapter Three.

Jancaitis & Junkins (1973) make use of local polynomials that must satisfy continuity and smoothness conditions along their boundaries and which are also submitted to further constraints 'which could be altitude or slope requirements'. They use as a typical local polynomial, the 15-term, fourth degree polynomial. The method of least-squares is used with Lagrangian multipliers, which must be computed together with the coefficients of all local polynomials.

(d) Other Patchwise Methods

The region of terrain coverage may be divided into a series of equal sized square or rectangular patches. Separate mathematical functions may then be generated to represent the surface within each patch. Methods of patchwise interpolation can be distinguished by whether the patches overlap or not. The difficulty with approximating functions is that discontinuities may occur along the boundaries, which make applications such as contouring, unsatisfactory. Overlapping patches will influence the surface in either two or four patches, thus having a smoothing effect on the resulting surface. Alternatively, these patches can be used to form a continuous surface over the common area of coverage.

The main advantage of patchwise methods over global methods is that low order polynomials can be used satisfactorily to describe each patch. Only a few unknowns have to be solved via simultaneous equations using least squares methods for each patch.

However, patchwise methods need greater organisation of data than with global methods, since the subdivision of the surface into patches needs to be carried out with care. If the data are poorly distributed towards the patch boundaries, the parameters or coefficients and thus accuracy will be affected. Surface patches have great potential, provided good mathematical approximations can be developed using low ordered terms. Factors affecting their efficiency include the number of points in each patch, and the size and number of patches. Provided suitable weighting functions are used to blend or smooth the patches together, the method can be efficient in terms of storage, point interpolation and other applications. The purpose of a blending function is to ensure that adjacent boundaries of the surface merge smoothly into one another. It is the mathematical form of the blending function which determines the continuity of the surface.

An example of this method is the surface averaging method, developed by Junkins et al (1973). The objective is to fit a surface over a regular grid of points by considering the surface as a set of individual sub-surfaces, each valid over a single grid cell, and each defined as a blend of four preliminary surfaces calculated for some of the surrounding cells. The weighted average of these surfaces tends to smooth the surface, so the size of the regions of validity has a major effect on the degree of smoothing. These local surface functions are fourth-degree polynomials, each requiring 25 coefficients. Alternatively, by using first degree polynomials as the weighting functions, the final surface patches would be second-degree polynomials, represented by 9 real coefficients, with little expected loss of accuracy between each. An alternative approach might be to store just the preliminary surfaces, and not the final weighted patches, but computation time for interpolation is significantly increased. In general, patchwise methods are computationally efficient, applicable to large data sets and very flexible, since any suitable function can be used for the preliminary surface. Junkins et al state *"if one least-squares approximation is good, then the average of four must be better"*. Tipper (1979) argues that this claim ignores the uniqueness of the optimal least-squares solution.

(e) Other Local Methods

Other methods of local approximation include the estimation of terrain elevations from neighbouring arbitrarily located points. Methods are distinguished by the interpolation technique used and the selection of points. The simplest selection technique is the nearest neighbour search, which locates a specified number of control points that are closest to the point being interpolated (Davis, 1986; Devereux, 1985). Unconstrained interpolation can be avoided by restricting the search to ensure that the control points are equally distributed about the interpolated point. Two modes of radial constraint are the quadrant and octant search, where a minimum number of control points are taken from each of the quadrants or octants. The procedure finds and tests more neighbours than in a simple

search, but the testing is a logic operation that does not greatly increase search time. The interpolation procedures usually involve averaging of heights weighted inversely by some function of their respective distances from the interpolated point. The weighting function actually assigns proportional weights, and expresses the relative influence of each control point. A widely used version assigns a function whose exact form depends upon the distance from the interpolated point and the most distant point used in the estimation, or in another variant, over the distance to the outer limits of the neighbourhood. The inverse distance-squared weighting function is then scaled so that it extends from one to zero over this distance. McClain (1974, 1976) addresses the problem of local estimation of arbitrary data points for applications such as contouring.

(f) Computer Aided Design (CAD) Methods

CAD was devised as a design tool in the aircraft, motor and shipbuilding industries. The theoretical basis of many of these techniques is sufficient to regard them as methods which generally use sets of local surface patches which, when blended together, create complex surface forms. The CAD approach is to design a model from scratch, either by amalgamating simple solid forms or by defining the object by its bounding surfaces, until it satisfies certain functional constraints. In contrast, terrain modelling involves developing a mathematical model which conforms, however approximately, to a specific landform. CAD methods can be incorporated as terrain modelling approximation techniques by organising the data into a network, such as a triangular or quadrilateral mesh, which is usually regular. The quadrilateral approach is compatible with most CAD methods, since the cell is used as the base for a single surface patch. It also reduces the surface representation problem to a standard bivariate interpolation or approximation. Tipper (1979) and Barnhill (1977) discuss some of these CAD methods. They are considered to be more of an interpolation technique than an approximation technique, in a similar manner to that of simultaneous patchwise, spline and continuous surface patch methods.

2.6 Chapter Conclusions

The range of terrain modelling techniques and DEM data structures is immense. This is due to the wide ranging needs of the users, who each have specific requirements for specific applications. These requirements may be efficiency (storage and time for retrieval), accuracy (approximating or interpolating model), flexibility for a range of applications, or a combination of these. Most of these models have been developed as an alternative to the regular grid DEM, due to its data redundancy and high storage costs. The models discussed overcome these problems, or have been designed specifically for particular applications.

This extensive literature review was undertaken to identify and classify methods of representing terrain and hence the advantages and suitability of the methods for generating terrain profiles, whilst offering substantial savings in storage over the regular grid DEM. The resulting classification links together methods with similar advantages and disadvantages, such that any model will be fairly representative of other models within that class. The structure of this classification is highlighted in Figure 2.2. The aim was to implement prototype terrain models from each of these representative classes and to examine in detail the suitability of each. It is not possible to identify, from the literature alone, the best data structure or modelling technique for any particular application. However, a number of promising alternatives have been identified. Terrain data models can be broadly classified into topological and mathematical structures. Topological methods are based on storing terrain features and the relationships that will adequately represent the surface, whilst mathematical models use a functional approach to approximate or interpolate the terrain.

Topological models store points regularly, semi-regularly or irregularly. The most common approach is to store regular DEMs, usually on a square or rectangular grid or lattice. However, it is faced with a series of constraints that have limited its use for digital terrain modellers. Since data are stored regularly, the data structure is not adaptive to the variability in the terrain, resulting in acute data redundancy. Despite this problem, the regular grid is the most popular DEM, due to its simplicity, flexibility, spatial addressing and time efficiency for retrieval operations, and the availability of terrain data sampled in a grid format. Because of these reasons, the regular grid DEM will act as the test model to which other models will be compared, for the purposes of this research. A fuller description of the performance of the regular grid DEM is given in Chapter Three. The use of data compression techniques also warranted further investigation, since methods such as Huffman coding can eliminate the redundancy within most data sets. This variable length code is assigned on the basis of some statistical criterion relating elevations, thus adapting itself to terrain variability. An investigation of the use of grid data compression is discussed in Chapter Five.

The most natural alternative to a regular grid is a semi-regular grid DEM. Some of the spatial attributes of the grid are retained, but it is adaptive to the variability in the terrain. For example, semi-regular DEMs can be regular profiles at irregular points (drop lines), or dense and sparse sub-grids of points (heterogeneous grids). Both methods overcome the problem of redundant data to some extent, but create problems in spatially indexing and storing points. The cost in storing these data structures may still be high, due to the overheads of the spatial addressing. However, the use of variable density grids has been fully examined in Chapter Four, together with the applicability of data compression techniques, such as Huffman coding in Chapter Five.

A method that combines the advantages of a variable resolution grid with mathematical modelling is the surface patch quadtree. Each local surface function is spatially indexed by a linear key that also determines its region of validity and spatial coordinates. The possible storage savings are great, and was thus chosen as a prototype model for further examination in Chapter Seven.

The next progression is from a semi-regular DEM to an irregular DEM. The major problem with these is being able to store many thousands of points, yet retaining the topological relationships in the data structure to facilitate data access and searching operations. Points can be indexed by point quadtrees or similar tessellation structures such as a grid; by the coordinate strings of contours and/or other terrain features such as ridges and channels; or by an irregular lattice based upon triangles or quadrilaterals. The success of such methods depends on the ability to locate data points, irrespective of the terrain features they represent. Polygonal, contour and similar DEMs tend to be inefficient, since entities are stored independently of one another, thus excessive search time may be required to locate points on a neighbouring contour or feature. Irregular networks such as the TIN incorporate the points of all terrain features together, as do overlayed grids or quadtrees. The benefits of using a triangulated irregular network and a local grid referencing scheme for irregular points are examined in Chapter Eight.

Mathematical representations of terrain have been in widespread use for over thirty years, mainly in the field of trend surface analysis by geologists. Methods can be distinguished on whether they approximate or interpolate the original data points. In general, for large area terrain modelling, approximation techniques are used, since they smooth or compress the data. These are usually based on least squares techniques, in which the residual errors are minimised. Interpolation techniques are not storage efficient for representative terrain models, and are generally used when the original elevations need to be preserved, such as for grid interpolation and accurate contouring.

In general, surface function approaches are not particularly applicable to arbitrarily large or global data sets and do not give consistent results in elevation and partial derivatives along boundaries. Also, surfaces that tend to interpolate the data exactly, give rise to an unrealistically rough surface. Local techniques such as patch methods mean that the benefits of each function can be used within the confines of a restricted region, so that the above problems are minimised, particularly when applied to grid data. The surface is then considered as a series of distinct local surfaces. The problem of discontinuities along adjacent surfaces is overcome by blending the surface patches together, but the overheads of doing this may detract from the large storage reductions that can be attained by storing each local function independently.

From the mathematical methods outlined, the use of polynomials has been consistently highlighted, both globally and locally for approximating and interpolating data. It has a good ability to adapt to almost any terrain surface, given specific requirements. The adaptability of local polynomial surface functions was examined in greater detail, the results of which are discussed in Chapter Six.

This review of terrain modelling techniques has highlighted methods which are worthy of further consideration and investigation (Chapters Three to Eight), for the purpose of acquiring comprehensive performance results. Prototype models for each of these were implemented and tested for the application of terrain profile generation and estimating radio path losses. As well as this, an in-depth analysis of the advantages and disadvantages of each method for the criteria outlined in Chapter One (Section 1.8) can be made.

Chapter Three

The Regular Grid DEM

3.1 Regular Grid Data Structure

The regular grid is the most common method of terrain representation, due to its simplicity, efficiency and availability of data in this format. The term grid implies a network of values arranged in a rectangular or square mesh and calculated in such a fashion that the values at the grid nodes (where a given row and column intersect) are accurate samples of the surface being modelled (McCullagh, 1988).

The density of measurements required to obtain a specific accuracy will depend on the variability of each terrain surface. Grids can be sampled regularly at any horizontal distance, depending on the accuracy required for the application and data acquisition technique. However, the most common sampling intervals are between 30 and 100 metres. The problem of model grid size has been addressed by McCullagh (1988), whilst Balce (1987) considers some criteria for point sampling and suggests a recommended sampling strategy. Several theoretical models have been developed in an attempt to determine the optimum number of points to acquire for a DEM. Kennie & McLaren (1988) state that the most common are those based on the analysis of either the power spectrum of the terrain using Fourier analysis, variograms of the terrain or self-similarity using fractal techniques (Ayeni, 1978; Frederiksen et al, 1985). These theoretical techniques are currently limited in their applicability although they show some potential in the determination of sampling strategies for photogrammetric mapping over large areas (Balce, 1987).

The data sets used in this study have been sampled at 50 metre intervals in both the x and y directions. This is the standard interval used in all the currently available data supplied by the O.S. from the 1:50,000 map series. These 20 kilometre x 20 kilometre grids consist of 401 measurements in both spatial directions (see Appendix A).

A fundamental requirement for application efficiency is to have the terrain data in a form which the computer can easily understand and access, such as an array. Each point's height is represented as the element of a two-dimensional matrix, such that the x and y coordinates need not be stored. Due to the regular nature of the data, the coordinates of any point can be calculated by a simple mapping relating coordinate position to storage position. It is generally accepted that the elevation represents the height at the grid node itself, rather than the height at the centre of the grid cell. Thus, by storing the coordinates of the first point and the

grid node interval, the elevation of any grid point can be found by a mapping of its coordinates into the indices of its matrix position. Retrieval operations are very efficient, since heights can be directly accessed from the array, without the need to search through the data.

3.2 Grid Interpolation Techniques

There are a large number of possible interpolation procedures, some of which have been described in Chapter Two. In general, the choice of an interpolation procedure is made on the basis of intuition, logical considerations and experience (Leberl, 1973). The interpolation methods can be distinguished as pointwise, global, or patchwise. Since regular grid DEMs can be created using these methods, it is just as feasible to use the same procedures for interpolating points from regular grid DEMs. Global methods are unsatisfactory for large amounts of data, whilst patchwise or piecewise functions can incur storage overheads in ensuring continuity and smoothness along the patch boundaries. However, in a regular grid DEM, pointwise and patchwise interpolation can be considered identical. Pointwise interpolation avoids problems of computer storage, since each point is interpolated independently, using only the surrounding subset of reference points. The coefficients of the interpolation function will vary from point to point, thus allowing more flexibility, at a cost of extra computation.

The most commonly used interpolation method for a regular grid DEM is polynomial interpolation. The general form of this equation for surface representation is:

$$h_i = a_{00} + a_{10}x + a_{01}y + a_{20}x^2 + a_{11}xy + a_{02}y^2 + a_{30}x^3 + a_{21}x^2y + a_{12}xy^2 + a_{03}y^3 + a_{31}x^3y + a_{22}x^2y^2 + a_{13}xy^3 + a_{32}x^3y^2 + a_{23}x^2y^3 + a_{33}x^3y^3 + \dots + a_{mn}x^m y^n, \quad \dots [1]$$

where h_i is the height of an individual point i ; x and y are the rectangular coordinates of the point i ; and $a_{00}, a_{10}, a_{01}, \dots, a_{mn}$ are the coefficients. Since the coordinates of each point are known, the values of the polynomial coefficients can be determined from the set of simultaneous equations which are set up, one for each point. For any given point with known coordinates (x,y) , the corresponding elevation can be determined by a substitution into this equation.

Interpolation within a rectangular grid will make use of some or all of the terms of the bicubic polynomial (the first 16 terms of eq. [1]). In matrix notation, this equation can be written as:

$$h = \begin{bmatrix} 1 & x & x^2 & x^3 \end{bmatrix} A \begin{bmatrix} 1 & y & y^2 & y^3 \end{bmatrix}^T \quad \text{or as} \quad h = x^T A y \quad \dots [2]$$

where A is the coefficient matrix that has a_{ij} as the element in row $i+1$ and column $j+1$, for $i, j = 0$ to 3 .

The 16-Term Bicubic Polynomial

This is the most sophisticated of grid interpolants, since the elevations and derivatives are reproduced exactly at the grid nodes and change continuously as the interpolating point crosses from one grid cell to another. The coefficients of the polynomial are derived from the elevations and three derivatives at each of the four grid nodes (16 values). These include the first derivatives h_x and h_y which express the slope of the surface in the x and y directions, and the mixed second derivative h_{xy} which is a tangent in a diagonal direction.

The coefficient matrix (A) is then calculated as follows. The components of the vector x (from equation [2]) and of its first derivative are computed for the two values of x at the boundaries of the grid cell (0 and 1 in local coordinates). The four resulting column vectors are transposed and placed below each other to form a 4×4 matrix X . A similar matrix is computed for Y . By choosing a local coordinate system, so that the origin is set to a corner (eg. bottom left) of the grid cell, the matrices X (and Y) can be stored in advance of the interpolation, and will be the same for all grid cells. The elevations and derivatives can be arranged as the elements of another 4×4 matrix H , such that equation [2] can be assembled as :

$$H = X A Y^T \quad \dots [3]$$

$$\text{where } X = Y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \end{bmatrix} \quad \text{and } H = \begin{bmatrix} H_{00} & H_{10} \\ H_{01} & H_{11} \end{bmatrix} \quad \text{for } H_{ij} = \begin{bmatrix} h & h_x \\ h_y & h_{xy} \end{bmatrix}$$

$$\text{The solution of [3] is then} \quad A = X^{-1} H (Y^{-1})^T \quad \dots [4]$$

This coefficient matrix (A) is formulated from [4], or alternatively the coefficients can be written as functions of the elevation and derivative data and calculated directly. These functions will be the same for any grid cell using the same local coordinate system.

For each two adjoining surfaces, the profile at their common boundary is a third degree polynomial with respect to either x or y , as the case may be. Each polynomial is completely determined by the two grid nodes at the ends of the boundary and the two first order derivatives or slopes in the direction of the boundary at these nodes. As a result, the two polynomials are identical and the total polynomial surface is continuous. Further, along a common boundary $x = \text{constant}$, h_x is in each local surface a third degree polynomial in y . These two polynomials are similarly completely determined by the two values of h_x and the two values of their derivatives h_{xy} at the nodes at the ends of the boundary. As a result, these two polynomials are identical. In other words, the two local surfaces have the same values of

h_x along their common boundary. The same reasoning applies to the values of h_y along a boundary $y = \text{constant}$. Consequently, the total surface is also smooth (Schut, 1976).

The derivatives can be determined by numerical differencing of the elevations in the grid (z). For any point (i,j), the derivatives in a localised square coordinate system are calculated as

$$h_x = \frac{z(i+1,j) - z(i-1,j)}{2}, \quad h_y = \frac{z(i,j+1) - z(i,j-1)}{2}, \quad h_{xy} = \frac{z(i+1,j+1) - z(i+1,j-1) - z(i-1,j+1) + z(i-1,j-1)}{4}$$

An alternative method of bicubic polynomials can be formulated from the elevations at the grid nodes only. In this instance, the 4×4 elevations of a surrounding grid cell can be used to form the matrix H in [4]. However, the total surface is continuous but not smooth at the boundaries.

The 12-Term Bicubic Polynomial

This interpolant is a 12-term incomplete bicubic polynomial formed from the four elevations and eight slopes of the four corner nodes, obtained from [2] by setting

$$a_{22} = a_{32} = a_{23} = a_{33} = 0$$

The total surface is continuous, but is smooth only at the grid nodes. Alternatively, the polynomial can be formed from the 12 nearest grid nodes. In both cases, the equations defining the coefficients can be computed and stored for a local coordinate system (between 0 and 1).

The Biquadratic Polynomial

This interpolant is derived from [2] by omitting the cubic terms $x^3, y^3, x^3y, xy^3, x^3y^2, x^2y^3$ and x^3y^3 to form a nine term polynomial (Petrie, 1987b). Usually, the term x^2y^2 is also omitted, such that the remaining eight parameters are computed by making the interpolant fit the heights at the middle of the grid sides, as well as the nodes (Schut, 1976). The total surface is continuous, but smoothness is not enforced along the boundaries. The elevations at the middle of the sides can be calculated from the heights of the four nearest points on that grid line, such as

$$h = -h_{-2} + 9h_{-1} + 9h_{+1} - h_{+2}$$

which is used by (Schut, 1976). If the nine term quadratic is favoured, an estimate of the elevation at the centre of the grid cell will also be required.

The Bilinear Polynomial

This is a four term polynomial of the form

$$h = a_{00} + a_{10}x + a_{01}y + a_{11}xy \quad \dots [5]$$

which will give a fit at the four corner nodes and interpolates linearly along the boundaries.

By using a local coordinate system (between 0 and 1), such as

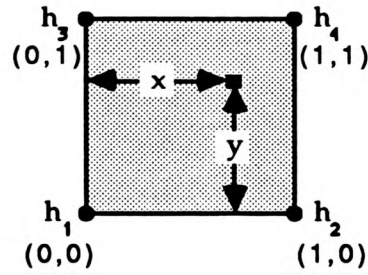


Figure 3.1 - Grid Cell in Local Coordinates.

where (x,y) are the local coordinates of the interpolate, the polynomial can be simplified to

$$h = h_1 + (h_2 - h_1)x + (h_3 - h_1)y + (h_1 - h_2 - h_3 + h_4)xy \quad \dots [6]$$

The Linear Polynomial

This is a three term polynomial that uses the three closest grid nodes. This is identical to surface fitting by plane triangles, such that

$$h = a_{00} + a_{10}x + a_{01}y \quad \text{or} \quad h = h_1 + (h_2 - h_1)x + (h_3 - h_1)y, \quad \dots [7]$$

for a point (x,y) in the triangle formed by points h_1 , h_2 and h_3 in Figure 3.1 above.

The Double Linear Polynomial

This is calculated as the arithmetic mean of two linear interpolations, since a point may lie in two triangles if both diagonals of the grid cell are considered (Leberl, 1973). This can be formalised as

$$h = (h_a + h_b)/2 \quad \text{where} \quad h_a = a_{00} + a_{10}x + a_{01}y \quad \text{and} \quad h_b = b_{00} + b_{10}x + b_{01}y. \quad \dots [8]$$

The above polynomial interpolation techniques have been used in a number of terrain modelling packages (Petrie, 1987a; Schut, 1976). They all interpolate or reproduce exactly the elevations at the grid nodes. The difference between them is the degree to which they approximate the terrain within each individual grid cell and the degree of smoothness along cell boundaries. Linear interpolation techniques do not permit any surface extremes, such as pits and peaks to be located anywhere but at the grid nodes themselves. Grid sampling must therefore be very fine, if these features are to be incorporated correctly. Alternatively, higher order interpolation techniques will allow extremes to be approximated within each grid cell, but not necessarily at the exact position. They will however give a better approximation of the terrain.

Since all the available terrain data is incorporated by such interpolating functions, it is

difficult to estimate their true ability to model the terrain, since there will be no 'real world' model to test their performance against. One can only surmise that the best approximation is one in which neighbouring nodes are taken into consideration to determine the local trend of the terrain and in which smoothness is maintained between grid cells. These more complex methods, such as the biquadratic and bicubic polynomials have been deemed sufficiently accurate to be used for producing contour maps (Jancaitis & Junkins, 1973; Junkins et al., 1973).

The relative performance of each of these six methods is illustrated below in an example. An 18x18 pixel, 50 metre regular square grid of terrain (subset of ST08 in Appendix A) is used to interpolate a larger grid of 69x69 pixels. Contours are then traced through these grid cells and plotted using straight line segments to connect points. The grid spacing is thus reduced from 50 metres to 12.5 metres. The original surface is shown below, with contours plotted at 5 metre intervals.

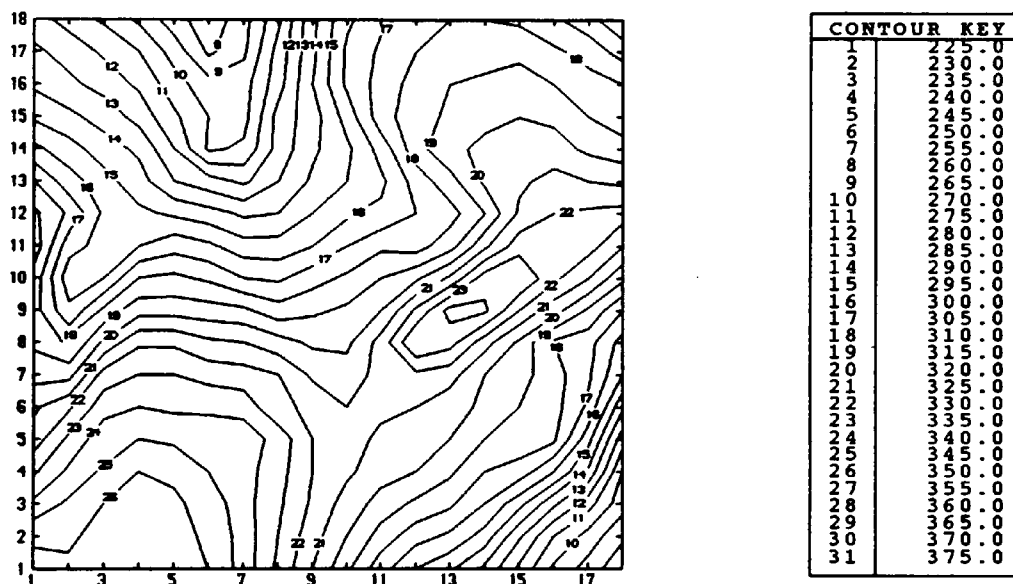


Figure 3.2 - The Original 50 metre 18x18 pixel grid of ST08 subset.

It can be seen that the contours are not smooth and exhibit sharp, unsightly changes in direction, or slope. This is due to the contours passing from one grid cell to another, with no enforced smoothing, since the gradient of the interpolated function changes discontinuously at the boundaries of each grid square. This contour map can therefore be considered unsuitable for most users. Polynomials could have been used to represent the points on the contours, to ensure smoothness for plotting, but since each polynomial is independent of one another, contours have a tendency to cross or overlap. However, by interpolating a denser grid using the methods described, these problems can be overcome, or at least reduced. A denser mesh, will ensure that the straight line segments joining contour points are shorter, so the contours appear smoother, whilst the higher order interpolation techniques, such as the bicubic polynomial, ensure a

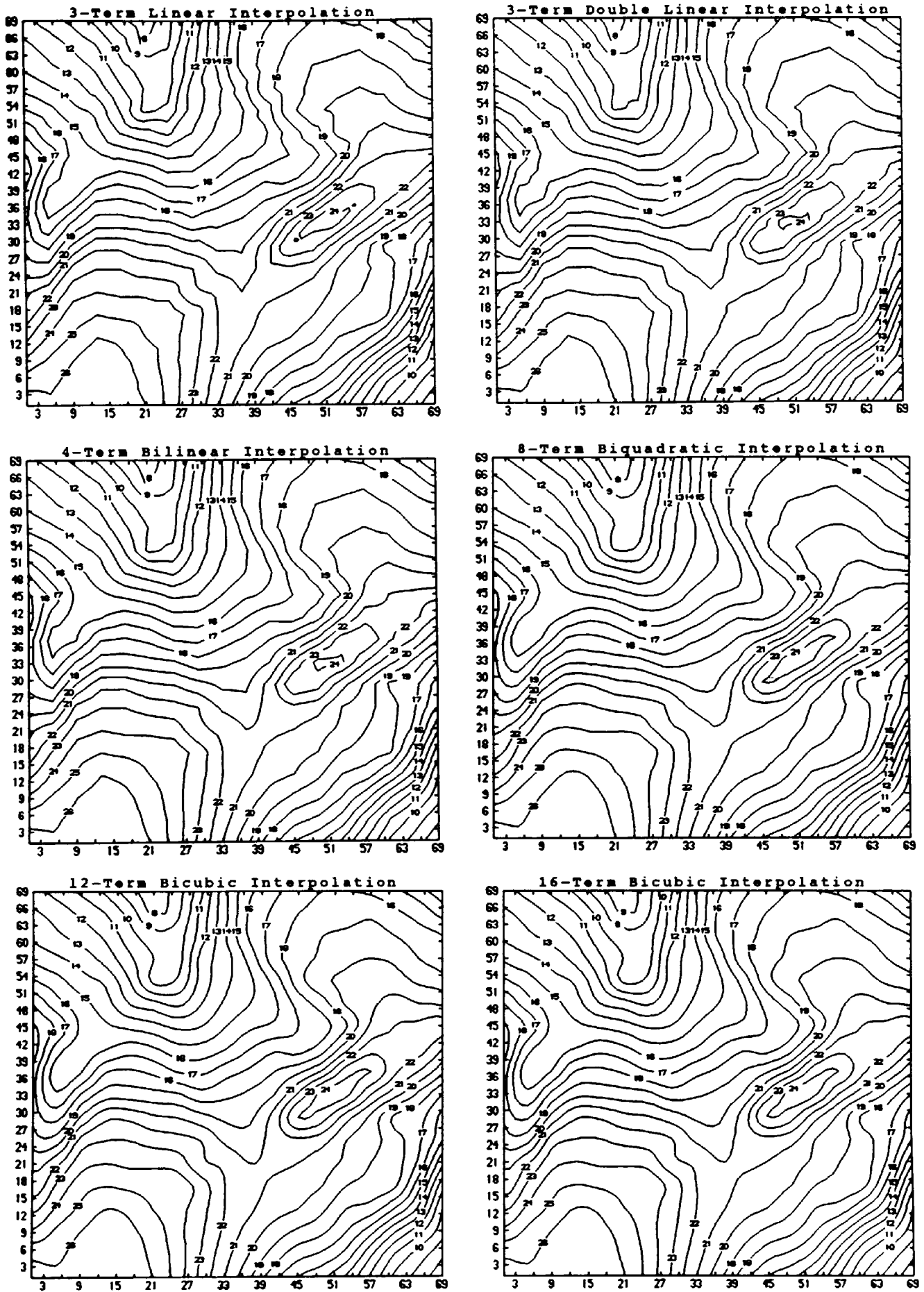


Figure 3.3 - (i) linear, (ii) double linear, (iii) bilinear, (iv) biquadratic, (v) 12-term bicubic and (vi) 16-term bicubic polynomial interpolated grid surfaces of 69x69 pixels.

degree of smoothness along the grid cell boundaries. The performances of the interpolation methods, compared to the 16-term bicubic polynomial are tabulated in Table 3.1, and the contour maps for the reconstructed 69x69 grids are shown in Figure 3.3.

Interpolation Method	Number of Terms	R.M.S.E	Largest Error
Linear	3	0.5364	2.8125
Double Linear	2x3	0.5350	3.2386
Bilinear	4	0.5320	2.7578
Biquadratic	8	0.1436	1.0018
Bicubic	12	0.0208	0.1377

Table 3.1 - Root Mean Square Errors and Largest Errors (in metres) of Interpolation Methods Compared to the 16-term Bicubic Polynomial.

From these contour maps (Figure 3.3), it can be seen that the higher the order of the interpolating polynomial, then the smoother the resulting surface will be. This is particularly true for the bicubic surfaces. The comparison of results for each surface with the full 16-term bicubic polynomial indicates that the overall performance of all methods is fairly good. The root mean square errors (RMSE) are generally low, but some individual errors of a few metres can occur for the linear surfaces. This illustrates the inflexible nature of interpolating methods that only allow local maxima and minima (peaks and pits) to occur at the grid nodes.

There are many other forms of grid interpolation which will produce satisfactory results. Jancaitis and Junkins (1973) discuss interpolation within a 4x4 sub-grid, where the centre cell contains the interpolate. This grid is then divided into four 3x3 sub-grids, to which biquadratic surface fits are made and the point interpolated. The four estimates are then averaged to produce the resulting elevation. The smooth nature of this interpolating scheme comes from the blending of the individual 3x3 interpolated values with a weighting function. Akima (1974a, 1974b) also uses a sub-grid to estimate points in his bivariate interpolation algorithm. The interpolation function is a bicubic polynomial, which requires the 16 values of the 4x4 sub-grid containing the interpolate and an additional 8 surrounding values, which are used to provide continuity and smoothness from cell to cell. An overview of these and other interpolation algorithms for digital terrain data are presented by Davis et al (1982).

Whilst higher order interpolation may produce better results for applications such as contouring (ie. where there is a need for smoothness and continuity to be maintained across the cell boundaries), the justification for its use solely as a point interpolant, such as for profiling, may be limited. The complexity of such methods may not be worth the limited improvement in

performance. Davis et al (1982) state that Akima's interpolant requires 78 floating point multiplications and 139 floating point additions for each interpolation of a point. Even if all the partial derivatives are stored beforehand, the interpolation still requires 22 and 64 operations respectively for multiplication and addition. Similarly, for profile interpolation, the standard biquadratic and bicubic methods discussed in this section may be considered too computationally expensive for the improved estimation. The performances of the polynomial interpolation methods derived from equation [1] are examined for elevation errors and computational efficiency in Section 3.3 and for radio path loss estimation in Section 3.4.

3.3 Grid Profile Interpolation

Each of the polynomial interpolation techniques discussed in Section 3.2 can be used to interpolate a profile through the terrain. However, an alternative method of interpolating a profile from a regular grid of heights is discussed by Edwards & Durkin (1969) and Dadson (1979), specifically for the purpose of predicting service areas for VHF mobile radio networks. This approach only interpolates points on the boundaries of grid cells and the diagonals that are intersected by the profile. The elevations at these points are linearly interpolated from the two grid nodes that form the cell boundary or diagonal. As a result, its error performance can be considered to be equivalent to that of the linear method discussed above. However, by interpolating the profile elevations only at the cell boundaries (ie. at the change in slope), no redundant information is calculated.

For the 500 metre grid advocated by Edwards & Durkin (1969), the average number of interpolations is five per kilometre, the worst case being two per kilometre when the radial coincides with a vertical or horizontal row of the grid. The profile points are interpolated at irregular horizontal intervals, so extra processing time will be required to calculate the intersections and organise the row, column and diagonally intersected elevations in profile sequence. In comparison with the other interpolation techniques, the three linear methods described in Section 3.2 will interpolate the same elevation values on the cell boundaries as Edwards & Durkin, but the double linear and bilinear methods may give more accurate results at the diagonals. Thus for comparative reasons, the results of the Edwards & Durkin method are not described here.

The effect of the height errors in grid interpolation for profile extraction was investigated for the O.S. grid reference ST08 50 metre grid DEM, together with the time efficiency of the various methods. A thousand randomly generated profiles were interpolated for each of the methods and compared to the results of the full 16-term bicubic polynomial. (It should be noted that these one thousand profiles are the same for all methods). Each profile radial is between

2 and 28 kilometres in length (average of 10.681 kms), interpolated at regular intervals of 50 metres (average of 214 points). The RMSE and largest elevation errors for all points of every profile are recorded in Table 3.2.

Interpolation Method	Number of Terms	R.M.S.E	Largest Error
Linear	3	0.490	9.33
Double Linear	2x3	0.399	6.47
Bilinear	4	0.400	6.48
Biquadratic	8	0.128	2.52
Bicubic	12	0.016	0.35

Table 3.2 - Elevation Errors (in metres) for the 1000 Profiles compared to the full 16-term Bicubic Polynomial.

The average RMSE for all interpolated elevations of the 1000 profiles is relatively low, but the largest single errors for the linear methods illustrate their failure to interpolate accurately within the grid cells. The biquadratic and bicubic polynomials give the best results, since the interpolated elevations are calculated from the trend of neighbouring elevations, not just the four corner nodes. The minimal differences between the 12 and 16-term bicubic polynomials show that there are no distinct advantages to be gained by using the former method, apart from slightly fewer calculations. The average interpolation times for the 1000 profiles are recorded below in Table 3.3. The timings are based upon implementation on a multiuser DEC VAX 8650 system and include a standard overhead of approximately 0.25 milliseconds for defining the interpolation points.

Interpolation Method	Number of Terms	Ave. Interp. Time (in ms)
Linear	3	0.275
Double Linear	2x3	0.321
Bilinear	4	0.277
Biquadratic	8	0.681
Bicubic	12	0.684
Bicubic	16	0.691

Table 3.3 - Average Profile Interpolation Time in milliseconds for the 1000 Profiles (Times include an average overhead of 0.25 ms, to define and store the points).

Table 3.3 shows that there is a difference in profile interpolation time for direct and indirect methods of calculation. The linear methods are directly applicable to the elevations at the

grid nodes of the cell, since they can be written as simple functions of these. However, higher order interpolation requires a series of coefficients to be calculated from some quite complex equations relating elevations and derivatives. The calculation of these coefficients from the defined equations is the most time consuming process of the profile interpolation. This accounts for greater than 60% of the total profile interpolation time, whilst for the linear methods the calculations are performed in 10% - 25% of total interpolation time. In these instances, the significant overhead is in calculating the coordinates of the points to be interpolated and storing the distances and elevations.

In general, interpolation from regular grid DEMs is very efficient since search time is minimal, due to the implicit mapping of grid data coordinates into matrix elements. Linear interpolation is approximately 2.5 times faster than higher order interpolation techniques, due to the direct nature of calculation from the grid nodes only. However, an overhead with double linear interpolation is ascertaining in which two of the four possible triangles the interpolate will lie. Higher order interpolation techniques are very similar computationally, so no distinct advantage can be gained by using biquadratic or 12-term bicubic interpolation instead of the full 16-term bicubic polynomial. The minimal saving in retrieval time does not warrant the loss of accuracy, however small. On the basis of these results, the choice of interpolant can be reduced to one of two approaches, namely a linear method or a higher order method. In terms of accuracy and corresponding profile generation time, the most promising methods for each class are bilinear and full bicubic interpolation.

3.4 Radio Path Losses

The radio path losses for the one thousand interpolated profiles of ST08 were calculated using the algorithm outlined in Section 1.7 of Chapter One. The path losses for the full 16-term bicubic polynomial interpolated profiles were again taken as the 'real world' model, against which the other methods were compared. The results of which are shown in Tables 3.4a - 3.4f for frequencies of 200, 400, 600, 900, 1400 and 1800 MHz.

Interpolation Method	R.M.S.E	Largest Error	Path Losses within		
			3dB	6dB	9dB
Linear	1.4873	19.2016	971	988	991
Double Linear	1.4883	19.2093	976	987	991
Bilinear	1.4764	19.2081	976	987	991
Biquadratic	0.8804	14.9532	988	996	997
Bicubic	0.5207	15.5139	998	999	999

Table 3.4a - Radio Path Loss Errors (in dBs) at 200 MHz

Interpolation Method	R.M.S.E	Largest Error	Path Losses within		
			3dB	6dB	9dB
Linear	1.2149	18.6404	967	990	999
Double Linear	1.0925	14.3320	972	989	999
Bilinear	1.0535	14.3577	973	991	999
Biquadratic	0.6448	10.0181	989	996	999
Bicubic	0.2999	8.6669	998	999	1000

Table 3.4b - Radio Path Loss Errors (in dBs) at 400 MHz

Interpolation Method	R.M.S.E	Largest Error	Path Losses within		
			3dB	6dB	9dB
Linear	1.2063	15.1571	964	995	1000
Double Linear	1.1168	11.9717	973	992	1000
Bilinear	1.0718	12.0438	974	993	1000
Biquadratic	0.5824	7.2841	990	998	1000
Bicubic	0.0323	0.4644	1000	1000	1000

Table 3.4c - Radio Path Loss Errors (in dBs) at 600 MHz

Interpolation Method	R.M.S.E	Largest Error	Path Losses within		
			3dB	6dB	9dB
Linear	1.1989	16.9146	974	991	996
Double Linear	1.1135	16.9301	978	991	997
Bilinear	1.1669	16.9301	977	990	996
Biquadratic	0.6902	7.4856	987	997	1000
Bicubic	0.1878	5.8362	999	1000	1000

Table 3.4d - Radio Path Loss Errors (in dBs) at 900 MHz

Interpolation Method	R.M.S.E	Largest Error	Path Losses within		
			3dB	6dB	9dB
Linear	1.3583	17.5978	965	985	994
Double Linear	1.2206	17.6133	977	990	995
Bilinear	1.2758	17.6133	975	989	994
Biquadratic	0.6529	8.4678	989	996	1000
Bicubic	0.2947	7.8928	998	999	1000

Table 3.4e - Radio Path Loss Errors (in dBs) at 1400 MHz

Interpolation Method	R.M.S.E	Largest Error	Path Losses within		
			3dB	6dB	9dB
Linear	1.5769	18.7097	960	981	992
Double Linear	1.4288	18.7251	969	987	992
Bilinear	1.4396	18.7252	966	987	992
Biquadratic	0.7309	10.9184	989	995	998
Bicubic	0.1484	3.9870	999	1000	1000

Table 3.4f - Radio Path Loss Errors (in dBs) at 1800 MHz

A significant conclusion obtained from these results is that the radio path loss algorithm is sensitive to certain profiles, ie. a small change in elevation may cause a large change in radio path loss. For example, for the 12-term bicubic polynomial, the largest elevation difference of any point in any of the 1000 profiles is no greater than 35 centimetres, yet for one profile a path loss error of over 15 dBs is observed at 200 MHz. On closer examination of errors such as this, it was observed that small elevation differences may cause an obstruction or diffraction edge to be missed/inserted or misplaced by up to a few points. Also at the lower frequency range (ie. 200 MHz), siting losses at the transmitter and receiver cause discrepancies for very small changes in elevation. These are not so critical at higher frequencies, since there is likely to be a small reflection loss instead. The 15 dB error in the example quoted above is due to the creation of such siting losses, despite it being a line-of-sight transmission path. These siting loss discrepancies are generally caused by very small hills in the vicinity of the transmitter or receiver.

In general, the majority of the errors are small ($< \pm 6$ dB), due to the fact that very often the increase/decrease in calculated path loss for missed/inserted or displaced obstructions may be unintentionally compensated for by an increase/decrease in magnitude of the diffraction losses of other obstructions. However, larger errors are likely to occur when a diffraction edge is missed or inserted, and an obstruction is misplaced by a few points. Some of these misplaced obstructions may also cause siting losses to arise, especially if they are near to the transmitter or receiver. Another noticeable feature of the algorithm is that the attenuation losses attributed to these occurrences will vary, depending on the radio frequency. The variation in RMSE with frequency is illustrated in Figure 3.4 below. This shows that the algorithm is likely to give the most consistent radio path loss estimates at between 400 and 900 MHz. The errors at 200 MHz are mostly due to the effect of minimal reflection and hence siting losses, which are not stable for small height errors. The linear methods are consistent with one another at all frequencies, whilst the RMSE for the biquadratic and 12-term bicubic are more easily influenced by the large errors of a small number of profiles.

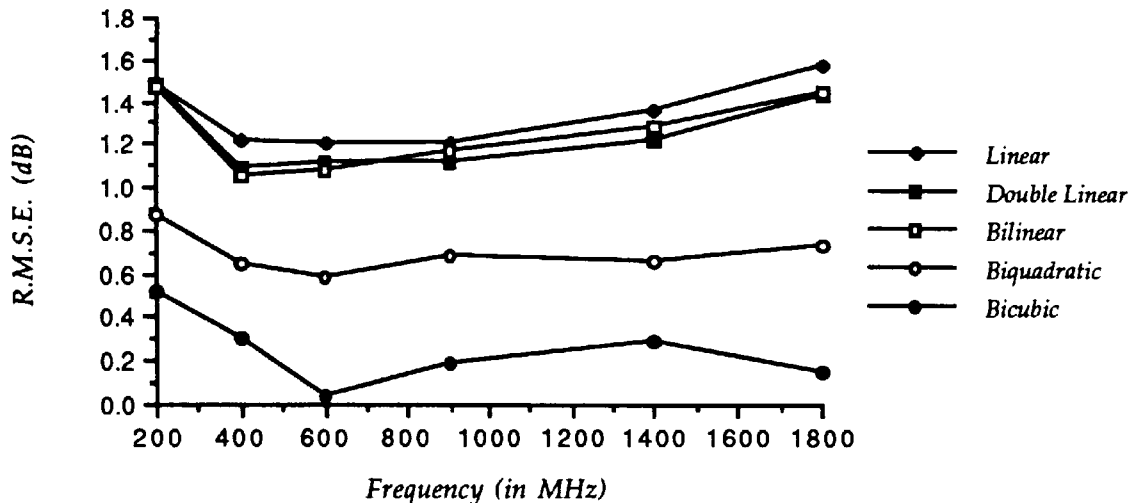


Figure 3.4 - R.M.S.E. of the Various Interpolation Methods with Frequency

The graph below (Figure 3.5) illustrates the number of profiles, out of the one thousand that are within a path loss error of ± 3 decibels. From this (and Tables 3.4a - 3.4f) it can be concluded that for linear interpolation techniques, up to 4% of profiles will be affected at the ± 3 dB tolerance level, and up to 1.3% of profiles for higher order interpolation. However, at the ± 6 dB tolerance level, no more than 1.9% of profiles will be affected for any interpolation method. This indicates that whilst the radio path loss algorithm is very sensitive to some small errors in elevation, these occurrences are not very common, since only 0.9% of profiles give path loss errors greater than ± 9 dBs.

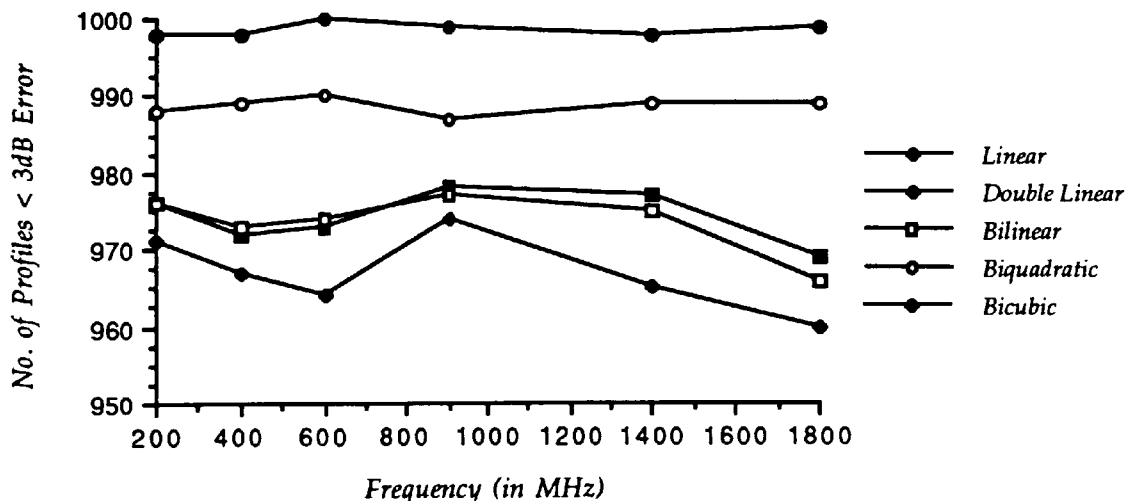


Figure 3.5 - Number of Profiles (Max. 1000) within a 3 dB Radio Path Loss Error

3.5 Chapter Conclusions

The use of various simple polynomial methods has highlighted the fact that interpolation

errors can occur in regular grids, simply because of the different interpretation that the methods make of the trend of the terrain within each grid cell. Linear methods assume that surface extrema (pits and peaks) can only occur at the grid nodes themselves, whilst higher order methods predict these within the grid cells from the neighbouring elevations. Despite having no easy way of determining the accuracy of these methods, it can be assumed that from the way that regular grids have been sampled, generated or interpolated, the non-linear techniques will give better estimates. In general, they model the characteristics of terrain more realistically, although linear methods may perform better for surface breaks, such as cliffs, provided the sampling is fine enough.

It has been shown that the use of linear interpolation can give errors of up to 9.5 metres (Table 3.2) within a 50 metre square grid cell for a random value (of O.S. grid reference ST08), when compared to the same bicubic interpolated value. This indicates that whenever possible, higher order interpolation should be used if accuracy is the prime consideration. In general, the difference between interpolation methods is small, with a maximum elevation RMSE of 0.5 metres (Table 3.2) and a corresponding maximum path loss RMSE of 1.6 dBs (Table 3.4f) for linear interpolation. However, the use of bicubic interpolation is approximately 2.5 times slower than linear and bilinear interpolation. This suggests that there may be an acceptable compromise between accuracy (elevation and path loss) and profile generation time. In this respect, a comparison of the results would suggest that a linear method would be favourable due to its time efficiency. For this reason, bilinear interpolation can be deemed the most satisfactory method of interpolation. The average elevation RMSE is 40 cms and its average path loss RMSE is 1.25 dBs for all frequencies. Profiles can be interpolated in under 0.3 milliseconds, of which less than 1.5% can be considered sensitive enough to create path loss errors in excess of ± 6 dBs.

The degree of accuracy will also depend on the grid cell size. Obviously, the denser the grid, then the more accurate the terrain representation will be, since more features can be incorporated in greater detail. In these instances the variation between interpolation methods will become smaller. However, as the grid size becomes finer, the overheads such as storage and computational efficiency become larger.

For radio path loss calculations, the use of a fine grid can be considered imperative for accurate results, since the algorithm is sensitive to small changes in elevation. In this work, a 50 metre grid is used, but Edwards & Durkin (1969) advocate the use of a 500 metre grid, since they consider it 'normally adequate' for the application of field strength trials. In their opinion a sparser grid may offer an acceptable compromise between storage costs and computational efficiency. The degree to which the grid cell size can be reduced, without significantly affecting the radio path loss results, was deemed worthy of further consideration and is

examined in Chapter Four.

Whilst regular grid DEMs may provide useful estimates of certain average quantities, they cannot accommodate changes in topographic texture conveniently. They are not adaptive to terrain variability, and as a result may cause excessive data redundancy. Mark (1978a), believes that they have not arisen from the need to model terrain, but rather from convenient programming and machine storage. However, the efficiency of the grid for applications such as contouring or profiling have outweighed these disadvantages for those users whom storage is not a problem. For the application of calculating path losses in a mobile communications network, storage is a major problem, and as such will outweigh the benefits of convenient programming and time efficiency. For applications such as these, alternatives to the regular grid are required.

Throughout this research, the regular grid is considered the 'real world' model to which other terrain models are compared, with interpolation performed bilinearly for efficiency. Methods are compared in terms of storage efficiency, elevation errors, profile retrieval efficiency and radio path loss errors. However, it should be assumed that up to 1.5% of profiles may be affected in terms of radio path loss error, due to the incongruous nature of the algorithm for some small height differences. The aim of this comparison is to determine the 'best' alternative to the regular grid.

Chapter Four

Regular and Variable Sub-Sampled Grid DEMs

4.1 Introduction

Grid methods record elevations at selected lattice points covering the region of interest. All grid schemes have the advantages of implicit coordinates that do not take up storage, and rapid retrieval of indexed data. In order to record accurately the majority of terrain features, the sampling density of the regular grid must be very fine throughout the model. However, in uniform regions, this sampling density of points may be superfluous for the character of the terrain, resulting in excessive storage of redundant points. One approach to overcoming this data redundancy problem associated with dense regular grid DEMs is to reduce the number of points required to represent the surface model. This chapter investigates two alternative grid techniques, which considerably decrease the storage requirements of dense grid DEMs, such as the 50 metre grid discussed in Chapter Three. These methods are applicable to other grid sampling densities, but the greatest storage savings can be realised when working from very fine grids. The techniques implemented are the use of sparser or sub-sampled grids and variable density grids at different resolutions.

4.2 Sub-Sampled Regular Grids

The results outlined in Chapter Three were all calculated using a dense, 50 metre regular grid of elevations. However, Edwards & Durkin (1969) and Dadson (1979) suggest the use of a 500 metre grid DEM in a mobile communications system for estimating attenuation losses between transmitters and receivers. The use of a sparser grid such as this will give considerably better storage costs and faster retrieval times for profile interpolation. The effect of sub-sampling dense regular grids was examined to determine the degree of data redundancy in the 50 metre grid and the effect on elevation errors and radio path attenuation. From this investigation, the optimum grid sampling density in terms of storage, profile retrieval time and radio path loss performance could be evaluated.

Since regular grid DEMs are not adaptive to terrain variability, the use of dense regular grids attempts to overcome this problem at a cost of introducing data redundancy in uniform areas of terrain. Hence, the use of sub-sampled grids is a step backwards in determining the effect of terrain variability on radio path loss attenuation at different grid resolutions. If the original grid is sampled densely, points can be simply eliminated from the grid, or alternatively the DEM could be generalised as a weighted average of the existing grid nodes (Loon, 1984). This is

based on the assumption that the original elevations have been measured or sampled accurately, and are not approximated estimates. The sub-sampled grid cell sizes examined in this study are 100 m, 150 m, 200 m and 250 m (Figure 4.1). Each of these were compared to the regular 50 metre grid, as discussed in Chapter Three.

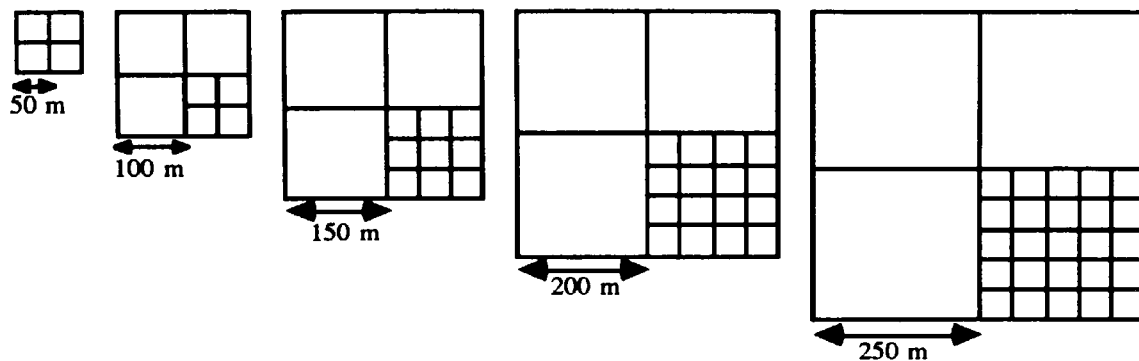


Figure 4.1 - Illustration of Grid Cell Sizes Compared to 50 Metre Grid.

From the above figure of 2x2 grid cells, some simple comparisons with the 50 metre grid can be made, concerning the number of grid cells and nodes. It can be seen that the reduction in grid nodes is not proportional to the reduction in grid cells, but will converge towards this level as the grid size increases. Since regular grid DEMs usually store the terrain elevations at the grid nodes, the degree of computer storage reduction for sub-sampled grids will depend upon the area of the surface to be modelled. For the standard test data sets used in this study (ie. 20 x 20 km DEMs of 401 x 401 points sampled at 50 metre intervals), the grid storage costs are illustrated in Table 4.1, together with the storage costs of the 2x2 grids in Figure 4.1 and the constant grid cell reduction factors. These results are based on a comparison with the original 50 m grid DEM.

Grid Cell Size	Grid Cell Storage Comparison		2x2 Grid Node Storage Comparison		Maximum Grid Node Storage Comparison	
50 m	1:1	100.00%	9:9	100.00%	401 ² : 401 ²	100.00%
100 m	4:1	25.00%	25:9	36.00%	401 ² : 201 ²	25.13%
150 m	9:1	11.11%	49:9	18.37%	400 ² : 134 ²	11.22%
200 m	16:1	6.25%	81:9	11.11%	401 ² : 101 ²	6.34%
250 m	25:1	4.00%	121:9	7.44%	401 ² : 81 ²	4.08%

Table 4.1 - Grid Cell and Node Storage Comparisons.
(N.B. Corresponding model at 150 metres relates to a 400x400 Grid at 50 metres).

The data storage costs of the sub-sampled grids above can therefore be approximated as:

$$\frac{1}{i^2} \times 100 \% \quad \text{where} \quad i = 1, 2, \dots, n = \frac{\text{Grid Cell Size}}{50 \text{ Metres}}$$

Hence, the data storage requirements of regular grid DEMs can be reduced significantly by a simple, yet crude elimination of points. The effect on the terrain elevation errors introduced by this data reduction was investigated for both O.S. grid reference ST06 and ST08 data sets (Appendix A). The points of the original 50 metre regular grid were bilinearly interpolated from each of these sub-sampled grids and the resulting errors recorded (Tables 4.2 and 4.3).

Grid Size	Abs.Ave. Error	RMSE	Stan. Dev.	% of Points within			Error Range
				± 5 m	± 10 m	± 15 m	
100 m	0.382	0.989	1.06	99.25	99.91	99.98	-32.00 to 26.50
150 m	0.713	1.612	1.77	97.78	99.66	99.93	-27.00 to 30.27
200 m	1.027	2.219	2.44	95.49	99.14	99.80	-50.75 to 41.25
250 m	1.386	2.878	3.20	92.79	98.29	99.48	-44.80 to 47.20

*Table 4.2 - Comparison of Sub-Sampled Grids with O.S. ST06 50m Grid
(Errors in metres)*

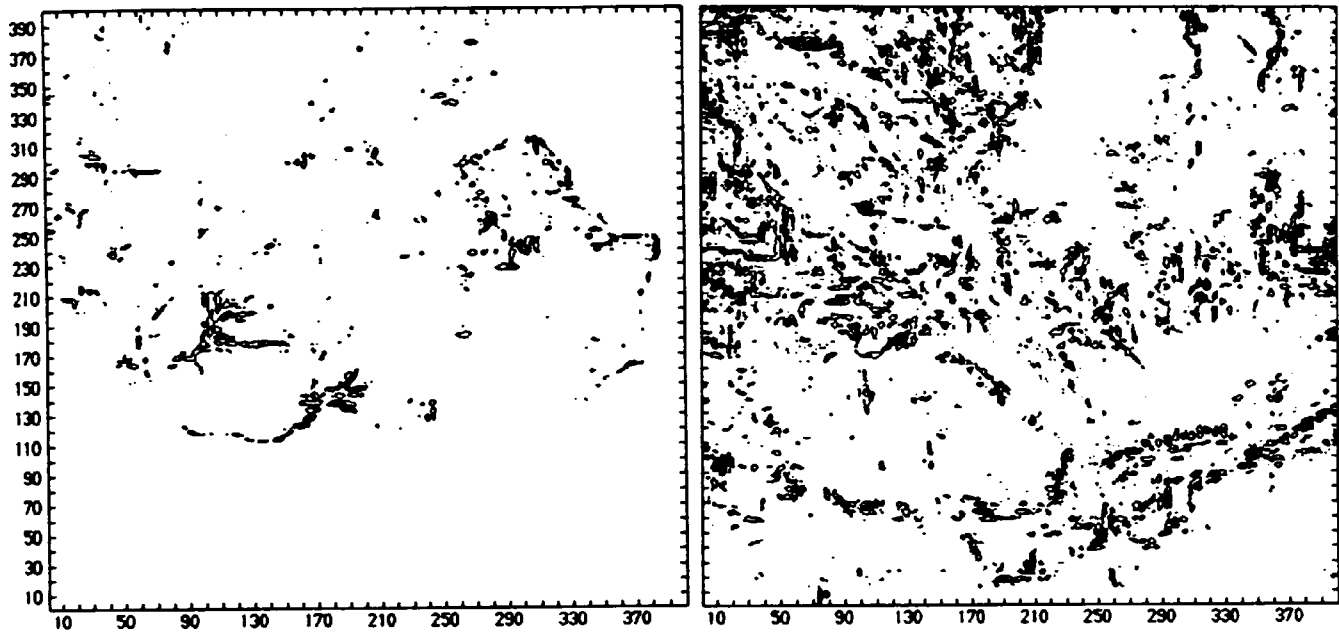
Grid Size	Abs.Ave. Error	RMSE	Stan. Dev.	% of Points within			Error Range
				± 5 m	± 10 m	± 15 m	
100 m	0.996	1.822	2.08	96.78	99.69	99.97	-21.50 to 30.00
150 m	1.914	3.112	3.69	90.58	98.29	99.67	-27.89 to 31.44
200 m	2.865	4.477	5.32	81.74	95.14	98.61	-37.81 to 38.00
250 m	3.999	6.125	7.35	72.61	90.28	96.33	-47.83 to 44.69

*Table 4.3 - Comparison of Sub-Sampled Grids with O.S. ST08 50m Grid
(Errors in metres)*

Further investigation of the location and cause of these errors concluded that the largest tend to occur in areas of steep gradients or slopes, such as valleys or cliffs. This accounts for the fact that the magnitude of the largest errors are generally greater for ST06 than ST08, due to the cliffs along the coastline causing sharp changes in elevation, despite ST08 representing more variable, higher relief. Plots of the error residuals for the 250 metre sub-sampled grids of ST06 and ST08 are shown in Figure 4.2.

These plots highlight the fact that a sub-sampled grid will always create excessive elevation errors, since they cannot accommodate terrain variability at lower resolutions. Even the smallest sub-sampled grids (ie. 100 metres) can create elevation errors in excess of 30 metres. Accurate surface representations can only be obtained with regular grid DEMs by using the densest sampling of points possible/available. However, the average elevation errors and RMS errors are generally quite small, especially for the 100 m and 150 m grids. In all the

sub-sampled grids illustrated above, there are certain areas within each surface where there are few or no significant errors. This suggests that the sub-sampled grid densities are satisfactory within certain limited regions. The results of weighting the sub-sampled grid nodes as a Gaussian function of the neighbouring elevations was examined as an alternative to the direct elimination of points, but was found to produce no significant increase in performance.



*Figure 4.2 - Residual Errors in 250m Sub-Sampled Grids for ST06 and ST08 .
(Errors shown at 10m intervals).*

The effect that the sub-sampling will have on profiling (for radio path loss estimation) was investigated further. As in Chapter Three, one thousand randomly generated profiles were used to test the average performance of the methods. Interpolation was performed bilinearly within the four grid nodes of a cell. Profile generation time was found to be directly proportional to the decrease in grid cell size (Table 4.4 and Figure 4.3), rather than the decrease in grid node storage requirements.

Grid Cell Size	Time (in ms)	Proportion of 50 m Grid Time
50 m	0.277	100.00 %
100 m	0.139	50.18 %
150 m	0.093	33.57 %
200 m	0.071	25.63 %
250 m	0.057	20.58 %

Table 4.4 - Sub-Sampled Grid Profile Interpolation Times

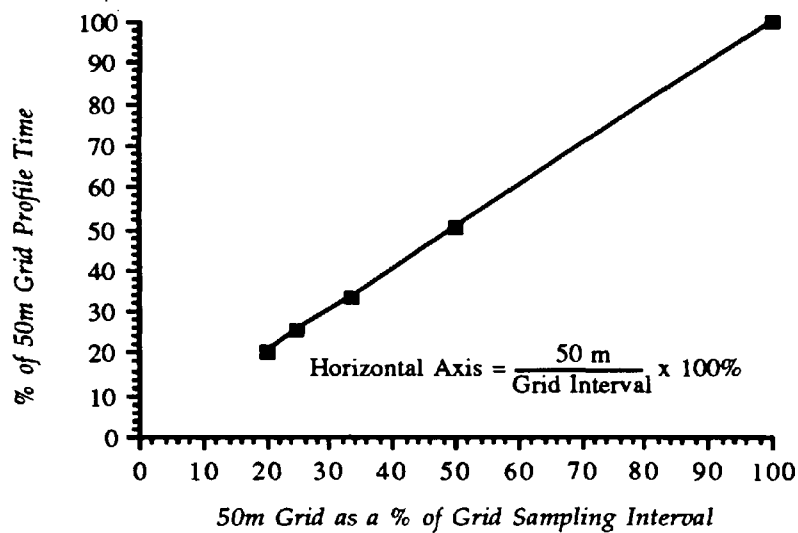


Figure 4.3 - Relationship between Grid Cell Width and Profile Interpolation Time

The errors associated with the 1000 test profiles which were bilinearly interpolated from these models were found to be consistent with the results in Tables 4.2 and 4.3, with the largest errors occurring in areas of steep slopes, although these were again relatively few in number. For ST06, which includes an area of sea (ie. Bristol Channel), profiles were not valid if both the transmitter and receiver were at sea level, since no obstructions would be located for any terrain modelling method, thus falsifying the true effective performance. Hence profiles were constrained such that either the transmitter or the receiver are above sea level.

The results for radio path attenuation at frequencies of 200 - 1800 MHz, in relation to the 1000 interpolated profiles of the original 50 metre grid are shown below in Tables 4.5a and 4.5b for ST06 and Tables 4.6a and 4.6b for ST08. These results indicate that to obtain at least 90% and 95% of profiles within an absolute error of ± 6 dB, averaged over all frequencies, the necessary sub-sampled grid resolutions would be 150 metres and 100 metres respectively for ST06 (11.22% and 25.13% of original grid storage). 95% of profiles within a path loss error of ± 6 dB cannot be attained for ST08, but approximately 93% are within this range at a grid resolution of 100 metres (25.13% of storage).

Cell Size	Frequency = 200 MHz				Frequency = 400 MHz				Frequency = 600 MHz			
	A.Av. Error	RMSE	No. of Profs ≤ 6 dB	No. of Profs ≤ 12 dB	A.Av. Error	RMSE	No. of Profs ≤ 6 dB	No. of Profs ≤ 12 dB	A.Av. Error	RMSE	No. of Profs ≤ 6 dB	No. of Profs ≤ 12 dB
100 m	0.985	1.902	978	997	0.993	2.071	963	999	0.981	2.120	964	997
150 m	1.513	2.598	956	994	1.619	3.108	929	989	1.596	3.023	936	985
200 m	1.947	3.078	931	994	2.023	3.655	909	978	2.022	3.670	907	977
250 m	2.436	3.617	905	989	2.598	4.250	859	970	2.667	4.343	851	971

Table 4.5a - ST06 Radio Path Loss Errors for Sub-Sampled Grids at 200, 400 & 600 MHz.

Cell Size	Frequency = 900 MHz				Frequency = 1400 MHz				Frequency = 1800 MHz			
	A.Av. Error	RMSE	No. of Profs ≤ 6dB 12dB		A.Av. Error	RMSE	No. of Profs ≤ 6dB 12dB		A.Av. Error	RMSE	No. of Profs ≤ 6dB 12dB	
100 m	1.044	2.277	953	995	1.165	2.581	947	991	1.238	2.768	947	986
150 m	1.737	3.276	914	982	1.975	3.756	905	975	2.100	4.029	899	967
200 m	2.237	4.023	881	973	2.530	4.587	865	965	2.689	4.925	858	952
250 m	3.015	4.886	815	960	3.393	5.552	793	938	3.602	5.934	785	921

Table 4.5b - ST06 Radio Path Loss Errors for Sub-Sampled Grids at 900, 1400 & 1800 MHz.

Cell Size	Frequency = 200 MHz				Frequency = 400 MHz				Frequency = 600 MHz			
	A.Av. Error	RMSE	No. of Profs ≤ 6dB 12dB		A.Av. Error	RMSE	No. of Profs ≤ 6dB 12dB		A.Av. Error	RMSE	No. of Profs ≤ 6dB 12dB	
100 m	1.636	3.236	941	980	1.525	3.044	936	987	1.632	3.371	933	982
150 m	2.473	4.452	898	967	2.651	4.568	868	966	2.560	4.558	879	968
200 m	3.354	5.393	827	950	3.597	5.799	794	944	3.403	5.466	806	949
250 m	4.156	6.367	782	931	4.493	6.928	740	903	4.232	6.490	754	922

Table 4.6a - ST08 Radio Path Loss Errors for Sub-Sampled Grids at 200, 400 & 600 MHz.

Cell Size	Frequency = 900 MHz				Frequency = 1400 MHz				Frequency = 1800 MHz			
	A.Av. Error	RMSE	No. of Profs ≤ 6dB 12dB		A.Av. Error	RMSE	No. of Profs ≤ 6dB 12dB		A.Av. Error	RMSE	No. of Profs ≤ 6dB 12dB	
100 m	1.554	3.313	932	984	1.641	3.631	930	978	1.733	3.868	927	973
150 m	2.519	4.504	876	969	2.687	4.913	872	951	2.837	5.231	862	943
200 m	3.344	5.343	806	951	3.593	5.857	800	929	3.785	6.228	795	918
250 m	4.242	6.511	757	912	4.594	7.142	746	884	4.838	7.581	735	874

Table 4.6b - ST08 Radio Path Loss Errors for Sub-Sampled Grids at 900, 1400 & 1800 MHz.

The results show that whilst the majority of profiles can be satisfactorily retrieved such that radio path loss is within ± 6 dB (ie. at least 78.5% of profiles for ST06 and 73.5% for ST08, at any grid sampling density and frequency), the overall performance of the grids is poor. The 100 metre grid is quite satisfactory, since at least 94.7% and 92.7% of profiles can be retrieved such that the radio path loss error will be within ± 6 dB for ST06 and ST08 respectively. However, sparser grids give considerably larger, unpredictable errors which can therefore be considered unsuitable for use in radio path loss calculations.

Figures 4.4 and 4.5 highlight the constant deterioration in performance at different grid sampling intervals. This suggests the use of even sparser grids will produce a similar, linear deterioration of results. Figure 4.4 shows for ST06, that as the grids become sparser, there is a wider error distribution between the frequencies at 200 and 1800 MHz, whilst for ST08 (Figure 4.5) the difference in path loss errors within this frequency range is fairly constant.

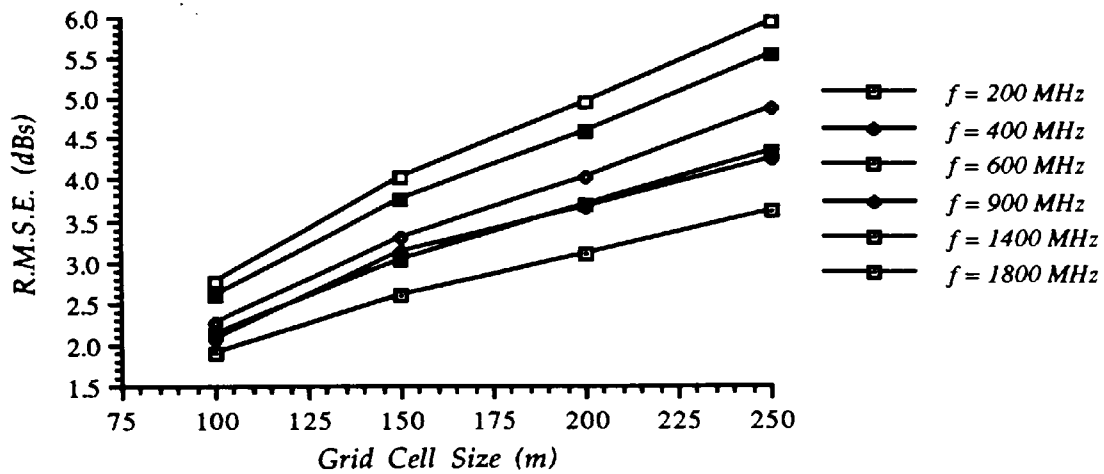


Figure 4.4 - R.M.S. Errors (in dBs) for ST06 at Frequencies of 200 - 1800 MHz.

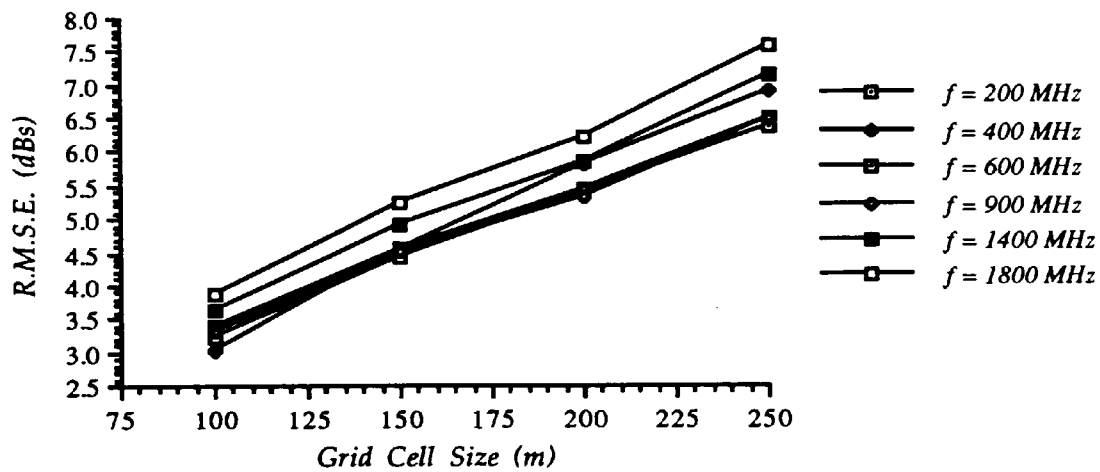


Figure 4.5 - R.M.S. Errors (in dBs) for ST08 at Frequencies of 200 - 1800 MHz.

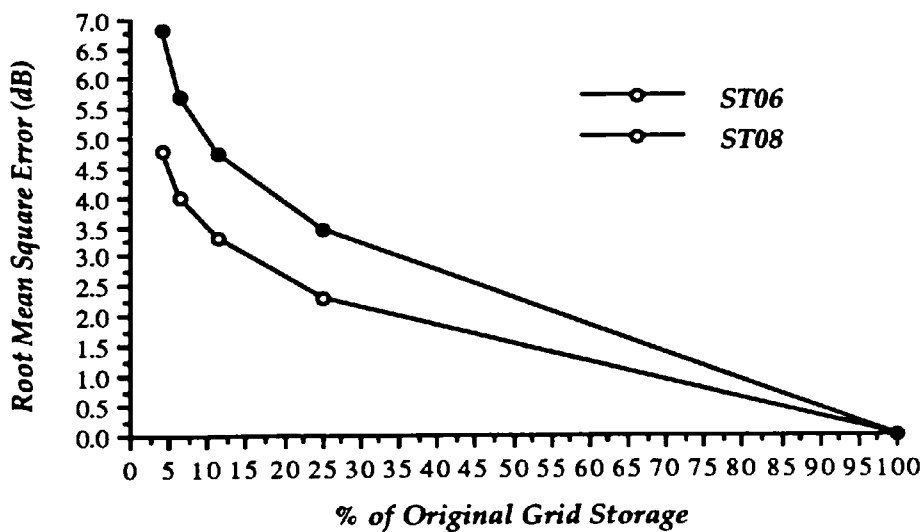


Figure 4.6 - R.M.S. Error (dB) of Radio Path Losses for Sub-Sampled Grid DEMs Averaged Over the Frequencies of 200, 400, 600, 900, 1400 & 1800 MHz.

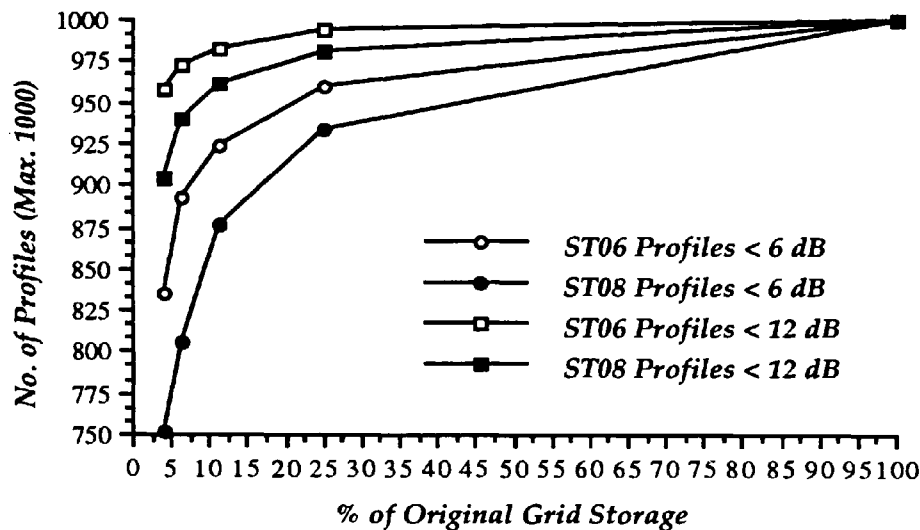


Figure 4.7 - No. of Sub-Sampled Grid Profiles Within Path Loss Error of ± 6 & ± 12 dBs Averaged Over the Frequencies of 200, 400, 600, 900, 1400 & 1800 MHz.

The performance of the sub-sampled grids in terms of storage, and averaged over all the tested frequencies (200 to 1800 MHz) are illustrated in Figures 4.6 (for RMSE) and 4.7 (for the number of profiles within ± 6 dB and ± 12 dB).

Sub-sampled grids remove the data redundancy of dense regular grids, but they are themselves unadaptable to terrain variability. The elimination of points in this systematic manner is simple, yet crude, resulting in the loss of critical points as well as redundant points. The method is unconstrained with no satisfactory criteria for point elimination, such as an inspection of surface characteristics or terrain roughness. The sampling of the original grid at a dense resolution causes data redundancy in ensuring that all terrain features are incorporated satisfactorily, but sub-sampling causes many of these important characteristics to be lost, despite removing the redundant points in some regions.

However, the use of the 100 metre sub-sampled grid highlights the fact that a large proportion of the data redundancy of dense regular grids can be removed, but up to 8% of profiles may give path loss errors in excess of ± 6 dBs. Whilst the use of sparser grids may cause some large path loss errors, at least 79.5% of profiles are satisfactory ($\leq \pm 6$ dB path loss error) for all tested sub-sampled grids (100 to 250 metres). This suggests that even at the sparsest resolution, specific regions of the surface are modelled satisfactorily. If sub-sampling can be constrained to be selective in certain regions, the benefits of data storage reduction can be realised without the introduction of uncontrollable elevation errors and their corresponding radio path loss errors.

Edwards & Durkin (1969) report that the use of a 500 metre grid is 'normally adequate' for

mobile radio network studies, whilst Dadson (1979) suggests that in terms of accuracy, a finer grid is not commensurate with the increased computation time required to carry out the additional calculations. In his study, Ackeret (1989) states that radio path profiling is sensitive to the resolution of the terrain elevation data in 'rough to very rough' terrain and generally insensitive in 'moderate to slightly rough' terrain. However, the categorisation of terrain roughness is a subjective process. Ackeret concludes that terrain model requirements, such as grid resolution, should be dependent upon each operational area. The results of this study supports this viewpoint, but in practice a 20 x 20 km terrain data set may exhibit several different terrain roughness tendencies (smooth, moderate, slightly rough, rough or very rough). As such, grid DEM resolution should not be constrained by such a rigid appraisal. Since terrain roughness is variable within a DEM, especially for large models, a viable alternative could be the application of variable resolution grids.

4.3 Variable Density Grid DEM

Sub-sampled regular grids will produce extensive storage savings for the corresponding region of interest and are computationally efficient, but cannot adapt to terrain variability. This results in significant elevation errors, which cannot be constrained or predetermined. A method of overcoming this problem is to use a variable grid or sub-grids with a flexible sampling resolution that adapts itself to the terrain roughness. Ideally, this variable grid should be sampled photogrammetrically, using a method such as progressive or composite sampling (Makarovic, 1973; 1975; 1977). However, they can also be derived from a dense regular grid DEM. Since it was shown in the previous section (4.2), that sparser grids are satisfactory in some regions of the terrain surface, this method attempts to identify where the sparser sub-grids can be used and its optimal sampling size.

The major problem with variable grids is determining the range of sampling densities and the optimal area of coverage. If the area at a specific sampling density is variable, then the data structure can be quite complex for access optimisation. This problem can be overcome by using a fixed patch size for each grid sampling density (Kostli & Wild, 1984). Hence the indexed data structure can be ordered such that it will provide direct access for retrieval operations, without the overhead of determining the region of validity for each sub-grid. Since the original test data sets (Appendix A) were sampled at 50 metre intervals, four grid sampling densities were considered in this study (at 50, 100, 200 and 400 metres), with a region of validity or patch size of 1600 metres² (ie. 400x400 metres). Each patch is represented by a homogeneous sampling of points. Hence sub-grids are of constant distribution throughout the model (ie. 2x2 grid nodes at a 400 metre resolution, 3x3 at 200 m, 5x5 at 100 m, and 9x9 at 50 m).

Each patch of the surface is examined in turn to determine the most suitable sampling density. This is accomplished by prescribing a maximum error tolerance for the interpolation of any original grid node, thus constraining the model to an accuracy standard. In the first instance, the 2x2 sub-grid at a 400 metre resolution are used to bilinearly interpolate the original 9x9 grid nodes. If at any stage an interpolated value differs from its original elevation by more than the prescribed threshold tolerance, this sub-grid is rejected. The process is repeated for the 200 and 100 metre resolution sub-grids, or until the error criterion is satisfied, in which case that particular sub-grid will be accepted in the final model.

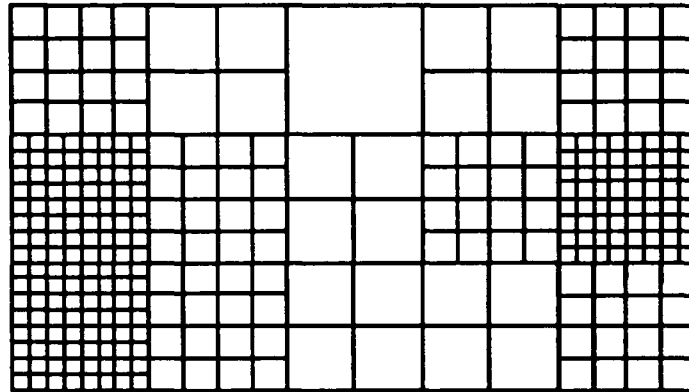


Figure 4.8 - Variable Grid of Different Sampling Densities for 5x3 Sub-Grids.

For example, in Figure 4.8 above, the 5x3 sub-grids represent a surface of 2000 m x 1200 m, which in the original 50 metre resolution grid would require 41x25 nodes (1025 points). The variable grid above consists of three 9x9 sub-grids at a resolution of 50 metres, six 5x5 sub-grids at 100 metres, five 3x3 sub-grids at 200 metres and one 2x2 sub-grid at 400 metres. Despite some duplication of points on the sub-grid boundaries, this data structure represents 442 points, a relative saving of approximately 57%.

The advantages of this method are clearly visible for application to grid DEMs. The major disadvantage of being unadaptable to terrain variability is overcome without creating too much data redundancy. The flexibility of the method allows terrain to be modelled locally at the most efficient resolution and constrains the maximum absolute error to a user-specified threshold.

Variable density grids were formulated for both ST06 and ST08 using error tolerances of up to 20 metres in increments of 2.5 metres. Tables 4.7 and 4.8 identify the percentages of sub-grids required to model the surfaces to within the specified maximum absolute error tolerances. These tables, together with Figure 4.9 also show the storage costs of the models as a percentage of original points in the regular 50 metre grid DEM.

Maximum Error Tol.	% of Patches Modelled by Grids of				% Points of 50 m Grid
	400 m	200 m	100 m	50 m	
2.5	43.76	7.52	16.72	32.00	50.824
5.0	55.28	15.96	17.08	11.68	27.154
7.5	67.44	16.28	11.04	5.24	17.449
10.0	77.48	13.48	6.56	2.48	12.439
12.5	84.64	9.84	4.64	0.88	9.600
15.0	89.60	7.24	2.76	0.40	8.203
17.5	93.16	5.16	1.44	0.24	7.414
20.0	95.08	4.16	0.56	0.20	6.999

Table 4.7 - Percentage of Grids at Various Resolutions for ST06 at Tolerances of up to 20 metres

Maximum Error Tol.	% of Patches Modelled by Grids of				% Points of 50 m Grid
	400 m	200 m	100 m	50 m	
2.5	1.92	2.68	15.60	79.80	107.588
5.0	7.24	16.32	31.24	45.20	72.157
7.5	18.68	25.12	33.96	22.24	46.113
10.0	31.52	27.04	31.60	9.84	30.570
12.5	41.72	29.80	25.00	3.48	20.968
15.0	49.96	31.16	17.68	1.20	15.929
17.5	58.16	30.28	11.08	0.48	12.829
20.0	66.36	26.12	7.16	0.36	11.073

Table 4.8 - Percentage of Grids at Various Resolutions for ST08 at Tolerances of up to 20 metres

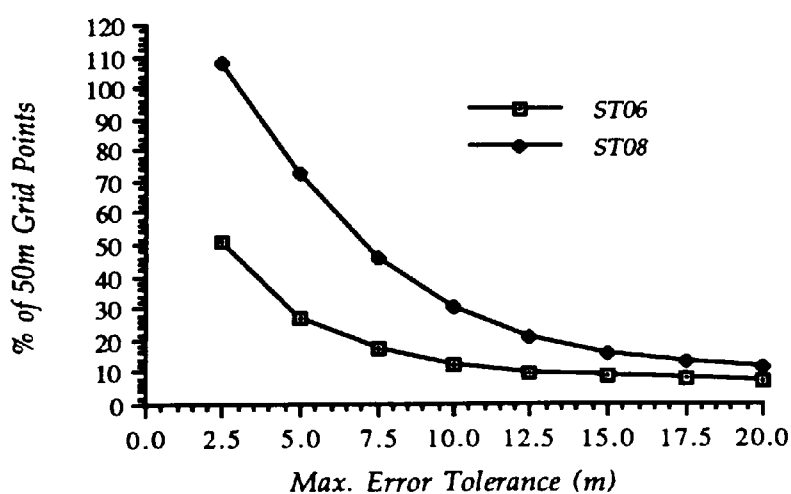


Figure 4.9 - Percentage Storage Requirements (of Original Points) with Error Tolerance.

It can be seen that the method has the potential to offer large savings in storage over dense regular grids, provided that the error tolerances are not too small. In Table 4.8, when the maximum error tolerance is 2.5 metres, the variable density grid will store more points than the original dense regular grid (107.59%). This is due to the large proportion of sub-grids at the original sampling interval of 50 metres (79.8%), thus incorporating data redundancy due to the duplication of elevations on the patch boundaries. However, as the error tolerance increases, the proportion of dense sub-grids decreases, thus allowing a more favourable distribution of sampling densities.

There is a noticeable difference in performance between ST06 (Table 4.7) and ST08 (Table 4.8), due to the former surface incorporating many sea level values. At these and in other less variable regions, the original dense regular grid exhibits extensive data redundancy. However, variable density grids can overcome this problem, by adapting the grid sampling density to terrain variability. Hence for ST06, the many sea level values can be represented by sub-grids of the sparsest sampling density, which in this case are 400 metre grids. If a DEM consists of many sea level values or relatively flat regions, then a sparser resolution could be used.

As the error tolerance increases, the storage requirements decrease towards a variable grid optimum of 6.22% of original points (ie. 50x50 sub-grids of 2x2 points sampled at 400 metres instead of 401x401 points sampled at 50 metres). The overheads of storing duplicate points along common boundaries of sub-grids is unavoidable, but the effect of this is generally minor and will contribute only a small proportion of the total storage cost. The consistent reduction in storage for the two data sets (Figure 4.9) would suggest that the relationship between storage and maximum error tolerance will behave in a similar manner for any surface. The terrain of data sets ST06 and ST08 are very different in nature, with the variability being representative of the two extremes of 50 metre regular grid DEMs for much of England and Wales. This would suggest that the storage costs for the majority of O.S. grid DEMs would exhibit a distribution within the range of ST06 (minimum) and ST08 (maximum) in Figure 4.9. However, there will be a few exceptions that will be above or below these extremes.

Variable density grids are illustrated in Figure 4.10 for ST06 and ST08 at the 10 metre maximum absolute error tolerance. These distribution maps can be compared to the original contour maps in Appendix A.

For retrieval operations, each of the 50x50 patches covered by the sub-grids are indexed by a value (1-4) indicating the point sampling density (400, 200, 100 or 50 metres). These values could be compressed into two bits, such that the additional storage overhead is only 625 bytes (ie. 50x50x2 bits, or < 0.2% of original grid storage). From this index, another index can be formulated to derive the reference position in the file data structure of each of the individual

sub-grids. At a cost of an additional one-off formulation of this index, point retrieval can be accomplished in a time equivalent to that of the original dense grid. However, if profile interpolation is accomplished by estimating elevations at the intersections with grid cell boundaries (Edwards & Durkin, 1969), the number of calculations will be fewer for a variable density grid, such that interpolation will be correspondingly faster.

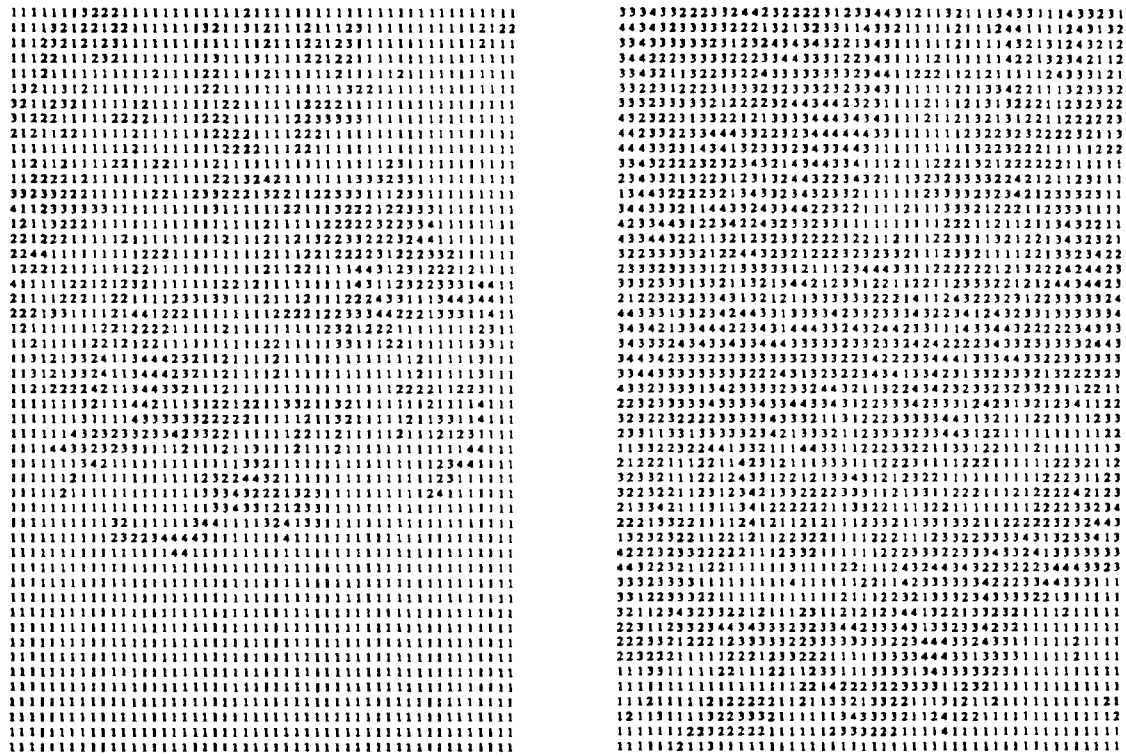


Figure 4.10 - Variable Density Grids for 50x50 Patches of ST06 and ST08 at 10m Error Tolerance.
(1 = 400x400m Grid; 2 = 200x200m Grid; 3 = 100x100m Grid; 4 = 50x50m Grid).

The elevation errors associated with the variable density grid for ST06 and ST08 are illustrated below in Tables 4.9 and 4.10. For every elevation in the original 401x401 grid, sampled at 50 metres, the corresponding point in the variable density grid is interpolated and the errors recorded. In each instance, the maximum error is constrained to the user-defined tolerance level, thus ensuring that the model has a consistent level of surface representation. The largest residual errors within the prescribed tolerance level occur in the regions of greatest variability, but these are uniformly distributed throughout the total surface model.

Max.Error Tolerance	Abs.Ave. Error	RMSE	Stan. Dev.	% of Points within		
				± 5 m	± 10 m	± 15 m
2.5 m	0.169	0.416	0.448	100.000	-	-
5.0 m	0.511	0.991	1.116	100.000	-	-
7.5 m	0.875	1.588	1.809	97.925	100.000	-
10.0 m	1.209	2.132	2.452	94.461	100.000	-
12.5 m	1.460	2.557	2.954	91.615	99.541	100.000
15.0 m	1.664	2.919	3.359	89.377	98.763	100.000
17.5 m	1.836	3.249	3.728	87.663	97.850	99.827
20.0 m	1.954	3.496	4.026	86.623	97.141	99.551

Table 4.9 - Comparison of Variable Density Grids with O.S. ST06 50m Grid (Errors in metres).

Max.Error Tolerance	Abs.Ave. Error	RMSE	Stan. Dev.	% of Points within		
				± 5 m	± 10 m	± 15 m
2.5 m	0.098	0.326	0.340	100.000	-	-
5.0 m	0.521	1.029	1.150	100.000	-	-
7.5 m	1.160	1.876	2.199	96.853	100.000	-
10.0 m	1.843	2.727	3.291	90.500	100.000	-
12.5 m	2.486	3.507	4.274	83.630	98.846	100.000
15.0 m	3.047	4.209	5.199	77.566	96.733	100.000
17.5 m	3.586	4.896	6.090	72.186	94.012	99.468
20.0 m	4.080	5.559	6.908	67.705	90.982	98.437

Table 4.10 - Comparison of Variable Density Grids with O.S. ST08 50m Grid (Errors in metres).

The relationship between the RMSE and maximum absolute error for each tolerance level is illustrated below in Figure 4.11 for both terrain data sets.

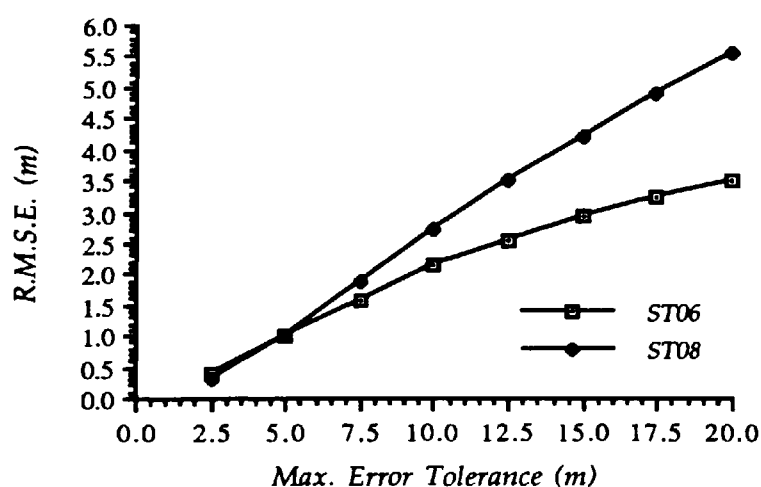


Figure 4.11 - Relationship between RMSE and Maximum Error Tolerance.

It can be seen that the relationship between RMSE and the tolerance of the constrained grid is fairly consistent, as is the relationship between storage and constrained tolerance (Figure 4.9). From the nature of the two terrain data sets, it can be surmised that for other surfaces, the RMSE is likely to be distributed between the minimum (ST06) and maximum (ST08) error bounds of Figure 4.11, for most of England and Wales.

For comparative consistency, bilinear profile interpolation was accomplished by estimating elevations at approximately 50 metre intervals, as in the original source grid, with a resulting generation time equivalent to that of the regular grid DEM. However, interpolation time is correspondingly faster if points are estimated only at the intersections with grid cell boundaries (ie. change in gradients), since fewer calculations are necessary. The radio path loss algorithm was applied to the 1000 test profiles for the variable density grids of ST06 (Tables 4.11a-b) and ST08 (Tables 4.12a-b) for constrained elevation tolerances of up to 20 metres.

Err. Tol.	Frequency = 200 MHz				Frequency = 400 MHz				Frequency = 600 MHz			
	A.Av. Error	RMSE	No. of Profs ≤ 6dB 12dB		A.Av. Error	RMSE	No. of Profs ≤ 6dB 12dB		A.Av. Error	RMSE	No. of Profs ≤ 6dB 12dB	
2.5 m	0.400	1.018	992	999	0.309	0.975	994	999	0.307	0.975	996	999
5.0 m	0.863	1.715	986	998	0.773	1.704	981	997	0.712	1.572	982	999
7.5 m	1.319	2.283	970	995	1.229	2.286	965	996	1.172	2.219	963	996
10.0 m	1.668	2.731	955	992	1.613	2.817	941	992	1.543	2.740	942	994
12.5 m	1.884	2.993	943	993	1.942	3.262	922	988	1.850	3.125	923	990
15.0 m	2.079	3.205	928	993	2.191	3.720	908	981	2.024	3.393	910	988
17.5 m	2.340	3.633	909	986	2.464	4.145	887	974	2.280	3.787	891	983
20.0 m	2.549	3.914	891	979	2.651	4.380	868	973	2.539	4.208	868	977

Table 4.11a - ST06 Radio Path Loss Errors for Variable Density Grids at 200, 400 & 600 MHz.

Err. Tol.	Frequency = 900 MHz				Frequency = 1400 MHz				Frequency = 1800 MHz			
	A.Av. Error	RMSE	No. of Profs ≤ 6dB 12dB		A.Av. Error	RMSE	No. of Profs ≤ 6dB 12dB		A.Av. Error	RMSE	No. of Profs ≤ 6dB 12dB	
2.5 m	0.334	1.004	991	1000	0.363	1.097	990	1000	0.381	1.146	989	1000
5.0 m	0.764	1.623	977	1000	0.851	1.824	973	999	0.897	1.931	971	998
7.5 m	1.263	2.365	951	999	1.394	2.645	941	996	1.472	2.812	940	994
10.0 m	1.683	2.975	924	995	1.878	3.342	912	986	1.988	3.560	908	980
12.5 m	2.019	3.372	906	993	2.261	3.808	892	978	2.394	4.057	885	969
15.0 m	2.238	3.714	892	988	2.511	4.234	879	969	2.668	4.528	869	960
17.5 m	2.524	4.152	866	979	2.846	4.748	849	959	3.024	4.748	842	948
20.0 m	2.808	4.554	838	972	3.172	5.192	823	948	3.375	5.564	814	934

Table 4.11b - ST06 Radio Path Loss Errors for Variable Density Grids at 900, 1400 & 1800 MHz.

Err. Tol.	Frequency = 200 MHz				Frequency = 400 MHz				Frequency = 600 MHz			
	A.Av. Error	RMSE	No. of Profs $\leq 6\text{dB}$	12dB	A.Av. Error	RMSE	No. of Profs $\leq 6\text{dB}$	12dB	A.Av. Error	RMSE	No. of Profs $\leq 6\text{dB}$	12dB
2.5 m	0.169	0.755	996	998	0.111	0.681	996	999	0.101	0.663	997	999
5.0 m	0.768	2.013	978	995	0.544	1.525	983	996	0.448	1.306	985	998
7.5 m	1.312	2.495	963	995	1.111	2.283	957	996	0.971	2.109	966	993
10.0 m	1.840	3.148	944	985	1.528	2.841	935	993	1.378	2.687	941	993
12.5 m	2.241	3.801	916	980	2.000	3.659	905	983	1.843	3.526	915	985
15.0 m	2.655	4.436	885	976	2.488	4.441	878	971	2.277	4.112	883	979
17.5 m	2.970	4.725	858	965	2.863	4.865	842	967	2.660	4.639	864	972
20.0 m	3.259	5.166	839	961	3.159	5.229	830	956	2.984	4.985	839	963

Table 4.12a - ST08 Radio Path Loss Errors for Variable Density Grids at 200, 400 & 600 MHz.

Err. Tol.	Frequency = 900 MHz				Frequency = 1400 MHz				Frequency = 1800 MHz			
	A.Av. Error	RMSE	No. of Profs $\leq 6\text{dB}$	12dB	A.Av. Error	RMSE	No. of Profs $\leq 6\text{dB}$	12dB	A.Av. Error	RMSE	No. of Profs $\leq 6\text{dB}$	12dB
2.5 m	0.109	0.714	997	999	0.120	0.797	997	999	0.126	0.846	997	999
5.0 m	0.473	1.397	982	998	0.515	1.551	982	997	0.542	1.647	982	996
7.5 m	0.995	2.194	953	995	1.076	2.438	951	993	1.142	2.617	947	991
10.0 m	1.412	2.831	933	991	1.528	3.153	929	987	1.621	3.385	920	982
12.5 m	1.851	3.561	914	982	1.992	3.958	910	971	2.104	4.224	904	965
15.0 m	2.272	4.151	881	976	2.466	4.558	871	964	2.606	4.860	866	958
17.5 m	2.688	4.673	852	968	2.905	5.110	840	950	3.065	5.443	836	945
20.0 m	3.054	5.082	823	961	3.310	5.554	811	946	3.485	5.900	802	933

Table 4.12b - ST08 Radio Path Loss Errors for Variable Density Grids at 900, 1400 & 1800 MHz.

Figures 4.12 (ST06) and 4.13 (ST08) below, show the radio path loss RMS errors for the tolerance models at frequencies of 200 to 1800 MHz.

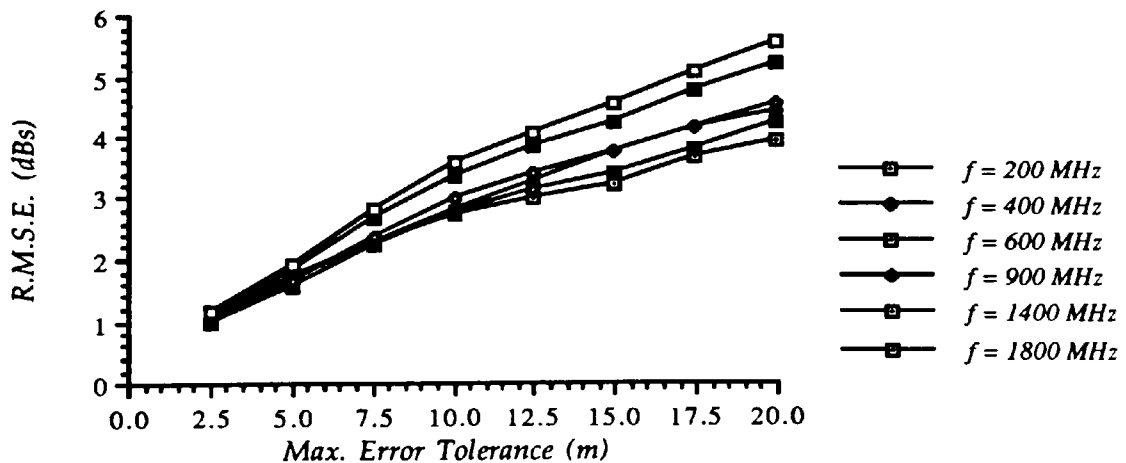


Figure 4.12 - R.M.S. Errors (in dBs) for ST06 at Frequencies of 200 - 1800 MHz.

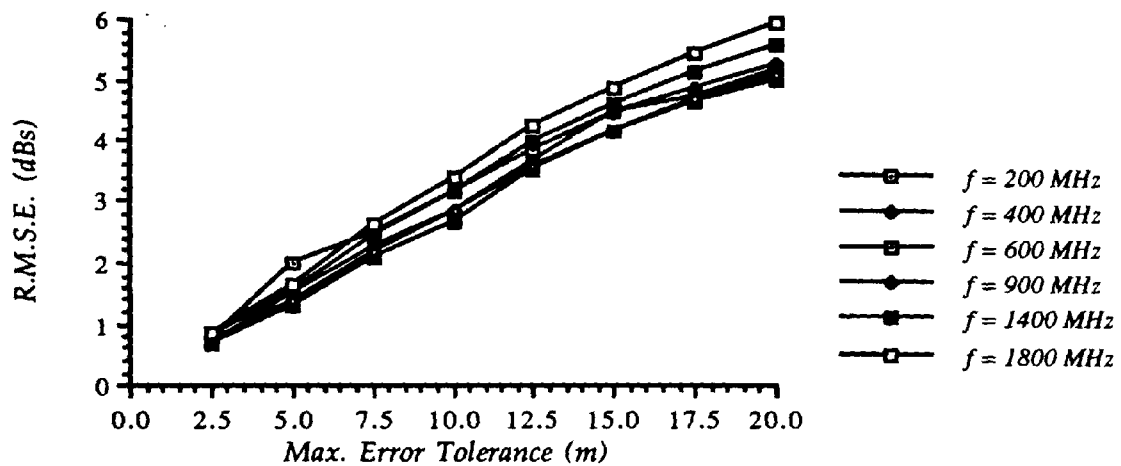


Figure 4.13 - R.M.S. Errors (in dBs) for ST08 at Frequencies of 200 - 1800 MHz.

The deterioration in radio path loss performance with increasing error tolerance (Figures 4.12 & 4.13) of the variable density grid is similar to the relationship for the sub-sampled grid DEMs (Figures 4.4 & 4.5). The error distribution for ST06 is wider than that of ST08 at frequencies of 200 to 1800 MHz, but both exhibit a linear deterioration in performance, the gradient of which is greater for ST08, as would be expected for more variable terrain. The radio path loss performance of the variable density grids averaged over all frequencies (200 to 1800 MHz) is illustrated below in Figures 4.14 & 4.15 in terms of RMS error and number of profiles within ± 6 and ± 12 dB.

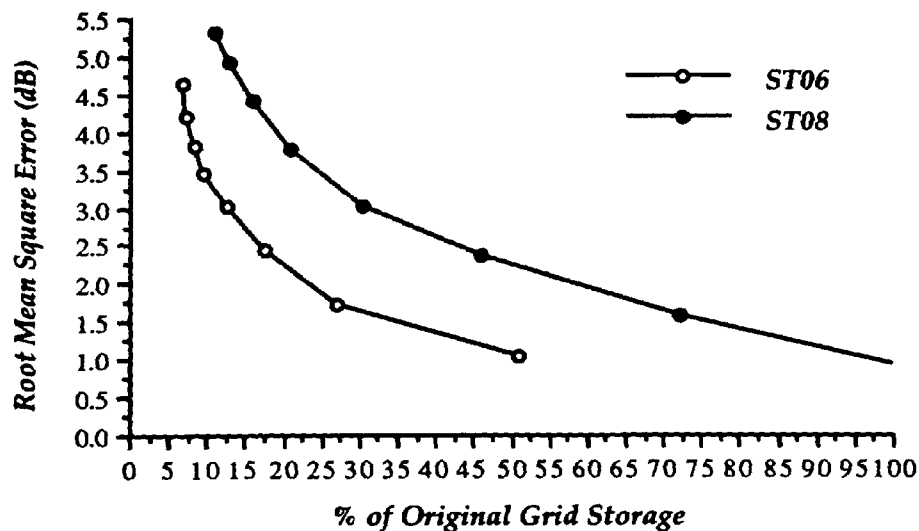


Figure 4.14 - R.M.S. Error (dB) of Radio Path Losses for Sub-Sampled Grid DEMs Averaged Over the Frequencies of 200, 400, 600, 900, 1400 & 1800 MHz.

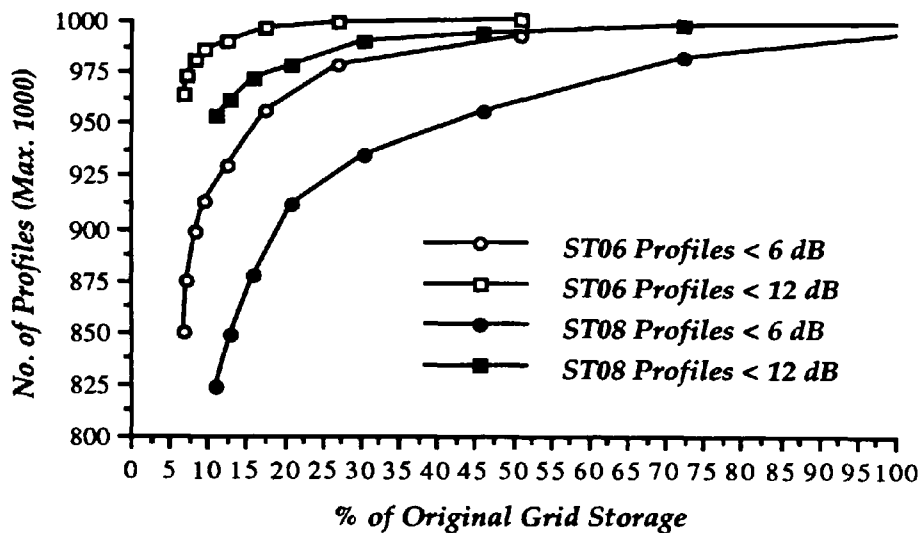


Figure 4.15 - No. of Sub-Sampled Grid Profiles Within Path Loss Error of ± 6 & ± 12 dBs Averaged Over the Frequencies of 200, 400, 600, 900, 1400 & 1800 MHz.

The overall results for radio path loss performance, indicates that to ensure at least 90% and 95% of profiles within an absolute error of 6 dB, averaged over all frequencies, the variable density grid would have to be formulated at elevation tolerances of 12.5 m and 7.5 m respectively for both ST06 and ST08. The corresponding equivalent storage costs as a percentage of the original 50 metre dense grid would be 9.6% and 17.45% for ST06 and 20.97% and 46.11% for ST08. This shows that considerable storage savings can be attained when compared to regular grid DEMs, whilst ensuring a good overall performance for estimating the radio path attenuation.

4.4 Chapter Conclusions

The two methods discussed in this chapter, namely the sub-sampled regular grid and the variable density grid, have both been assiduously examined for the effect of grid resolution on terrain modelling for path loss attenuation. Both methods can offer substantial storage savings over dense regular grid DEMs.

It has been shown (Section 4.2) that grid resolution can have a significant effect on the performance of radio attenuation calculations. Whilst the magnitude of this effect is dependent upon the characteristics of the terrain (ie. roughness) and degree of sub-sampling, it can be concluded that if a regular grid DEM is to be used, then the choice of sampling resolution should be as dense as possible for greatest accuracy. However, this is not the ideal answer, since an alternative to the dense regular grid DEM is sought.

The results for the sub-sampled regular grid have shown that even in the worst stated case, at a grid resolution of 250 metres (storage reduction = 96%), between 73% and 90% of profiles may be retrieved to within a corresponding radio path loss error of 6 dB. Whilst this is unacceptable as a viable alternative to the dense regular grid, it demonstrates that sparser grids are acceptable in certain instances. However, it is the determination of 'which instances' that required further examination. The difference in path loss results for ST06 and ST08 (Tables 4.5 and 4.6), especially at the lower frequencies, suggests that terrain roughness is a major contributory factor. In areas of high relief or variability, the systematic elimination of points will cause important surface features to be lost, thus creating profile and radio path loss errors. Since the nature of terrain is variable, a uniform grid will only suffice if the resolution is dense, hence the alternative approach is a variable density grid.

The variable density grid is a surface modelling technique that is adaptable to the variability of any terrain. The method constrains maximum error to a user-specified tolerance and significantly reduces the storage requirements, when compared to a regular grid DEM. Provided that an index to grid sampling density is maintained, retrieval operations for bilinear point interpolation can be performed in the same time as for a uniform grid. Other applications may be performed in a correspondingly faster time. The effect of elevation errors on radio path loss attenuation produces a linear deterioration in performance with tolerance, the gradient of which depends upon the nature of the terrain. 95% of profiles can be retrieved to within a corresponding 6 dB radio path loss error, whilst producing storage savings of between 53.89 and 82.55%.

Chapter Five

Compression Techniques for Grid DEMs

5.1 Introduction

The amount of data in a regular grid DEM determines more or less the accuracy with which the earth's surface is sampled. Whilst the regular grid is advantageous for automatic data acquisition, program and application simplicity, it is less suited for permanent storage, especially if it forms part of a geographic information system (GIS). Regular grid DEMs are often stored as two-dimensional matrices of two-byte (16 bit) elevations. However, the choice of storage unit size is often arbitrary, without regard to how much data is sufficient to represent what is known (Dutton, 1983). Using two-byte values, the 65,536 possible classes are rarely utilised, since the accuracy to which the terrain can be sampled cannot be guaranteed from data acquisition. Since the maximum difference in the earth's relief above sea level is about 8850 metres (elevation of Mount Everest), a complete grid DEM could represent elevations to within a vertical resolution of 13.5 centimetres, which is beyond the accuracy scope of most data acquisition techniques.

Hence, grid DEMs are stored to within the vertical resolution defined at the acquisition stage, a typical value for which may be one metre, as with the O.S. grid DEMs. For these grids, the feasibility of using a smaller resolution would not be consistent with the accuracy of the derived interpolation. Alternatively, the DEMs of the Institute of Hydrology (Morris & Flavin, 1990) are currently being derived in collaboration with the O.S. to within a vertical resolution of 10 centimetres. Since typical grid DEMs of the United Kingdom have a variation in relief of no more than 1000 metres, ten bits are required to represent the elevation classes at a vertical resolution of one metre ($2^{10} = 1024 > 1000$) and 14 bits at 0.1 metres ($2^{14} = 16,384 > 10,000$). Hence for two-byte integer values, the elevations for both representations would incorporate 37.5% and 12.5% storage unit data redundancy respectively, or 6.25% for each unused bit. For O.S. ST06 and ST08 (Appendix A), the relief ranges are 135 and 451 metres respectively, at a vertical resolution of one metre. Hence, the elevations of ST06 could be represented as 8-bit values (50% redundancy) and 9-bit values for ST08 (43.75% redundancy). This illustrates another form of data redundancy that is inherent within regular grid DEMs.

Data compression is an approach at overcoming this form of data redundancy, by making fuller use of the storage units that represent the grid elevations. Spatial data compression can be logical, physical, information retaining or information reducing (Clarke, 1985). Logical data compression may include changing the coordinates to a local origin, or using a different file

structure or digital representation, whilst physical data compression involves an alteration in the logical data structure to reduce the storage requirements.

This chapter investigates some techniques which store all the points of the original grid in a form which is less explicit than the regular grid, but which is free of redundancy. These methods include differential altitude grids (encoded elevation differences), and the statistical encoding (ie. Huffman coding) of these differences. These data compression techniques are either information retaining, which allows full reconstruction of the original grid after compression, or information reducing, which preserve a generalised version of the grid DEM. In both cases, the form of data compression is obtained by predictive coding. For most data sets, the variance of the differences obtained in some predictive manner is less than the variance of the original data, resulting in a more efficient quantisation with fewer levels and hence greater compression.

5.2 Differential Altitude Grids

One of the simplest error-free data compression algorithms (both conceptually and in terms of implementation difficulty) is the differential pulse code modulation (DPCM) technique (Jancaitis, 1978) or differential altitude grid. This technique simply stores the difference in elevations between neighbouring grid nodes, rather than the elevations themselves. The magnitude of numbers encountered is reduced, thus requiring fewer bits for representation, whilst maintaining the original grid elevations. For example, a grid of elevations can be represented by its first value (bottom left), plus the differences from its previous elevation in a continuous sequence :

Grid Elevations					Grid Differences					Direction
308	297	283	275	277	+13	-11	-14	-8	+2	
295	281	275	268	267	+14	+6	+7	+1	+6	
267	265	261	260	261	+8	-2	-4	-1	+1	
259	259	254	252	253	0	+5	+2	-1	-2	
256	253	246	250	255	256	-3	-7	+4	+5	

Figure 5.1 - Illustration of Differential Altitude Encoding of 5x5 Grid.

The method was applied to the 401x401 grid of elevations (sampled at 50 metre intervals) for ST06 and ST08 and the range of elevation differences recorded. For ST06 (altitude range of 135 metres or minimum of 8 bits of storage per elevation), the range of differences is -42 metres to +33 metres, whilst for ST08 (altitude range of 451 metres or 9 bits of storage per elevation), the range of differences is ± 56 metres. In both cases the difference range can be represented by a fixed seven bit code in the range of -64 metres to +63 metres, since $2^7 = 128$ values. Therefore, in

terms of information content, the storage savings are minimal (12.5% for ST06 and 22.2% for ST08). These savings increase to 56.25% when compared to the 16-bit regular grid DEMs.

An alternative approach which provides a more compact form of storage is to specify each difference between successive grid nodes as a fixed elevation difference (Δ), whose value is determined from local heights. This form of relative coding is known as delta modulation (Lynch, 1985), such that the difference between a given node and its neighbour is quantised into one of two levels. If the difference is positive, plus Δ is coded, whereas if the difference is negative, minus Δ is coded. The important feature of delta modulation is that it allows only two possible levels to be coded. Because of this, it is sometimes called a '1-bit' system.

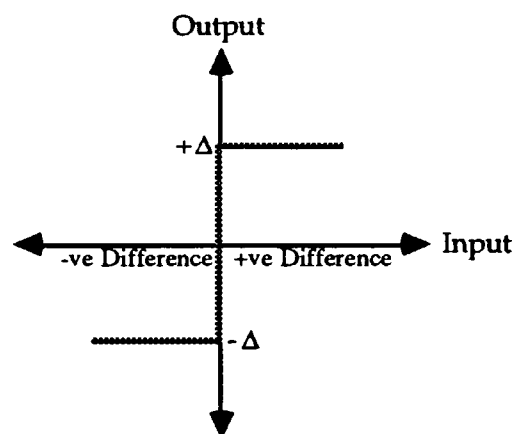


Figure 5.2 - Delta Modulation Quantisation.

Boehm (1967) uses this 1-bit delta modulation quantisation to represent 37x37 grid DEMs in a standard unit of storage called a microgrid, which consists of 39 words, each of 36 bits. However, the reconstructed waveform of a one bit code may not keep up with the original waveform (actual surface), since it restricts the representation to a monotone increase or decrease in step with no variation. This is a basic limitation of delta modulation, and is called 'slope overload'. Another limitation of delta modulation is when the relative differences are close to zero, resulting in positive or negative pulses in some random pattern. The effect of slope overload can be minimised by simply increasing the step size, but this may increase the effect of granular noise (variation above and below the waveform). By extending the code to two bits, this granular noise can be significantly reduced by adapting the modulation to a step increase, decrease or no change.

A terrain model was developed on this basis by defining two-bit differential codes to be valid over fixed size patches. The 401x401 grid DEMs were divided into 80x80 local surfaces of 5x5 nodes. The grid nodes are assumed to represent elevations at the centre of the grid cells, thus preventing elevations from being approximated twice in the surface model (ie. on the boundaries of two or more patches). However, the interpolation of points lying in the boundary

region of two patches will require both local surfaces to be accessed. This overhead in retrieval time is offset by a storage saving of 36% (ie. the cost of storing 100x100 patches instead of 80x80 patches). For each local surface, the first elevation at [1,1] is used as the base height from which other elevations are approximated. The differential altitude or correction height (Δ) that is applied to the nodes is calculated as the value that will give the best overall performance for the local surface. On examination, good approximations for this were found to be a quarter or a fifth of the local surface relief range. The first elevation is stored in nine bits, whilst the differential altitude is stored in seven bits, two of which represent the fraction part to the nearest 0.25 metres. The codes representing the application of the differential altitude are stored as four 2-bit values (one byte) for the heights in the first column, and five groups of four codes (five bytes) for the row corrections. The local surface of 5x5 grid nodes are therefore represented as:

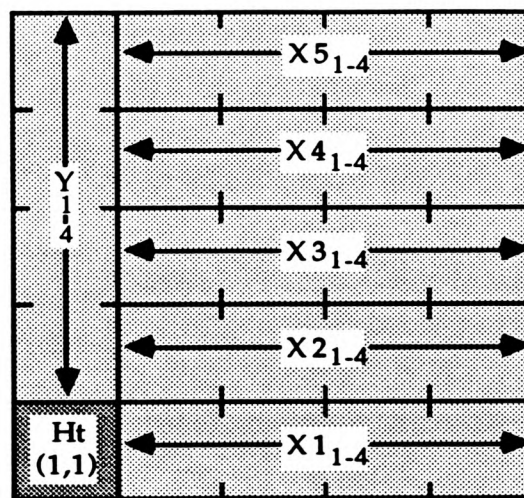


Figure 5.3 - Representation of 5x5 Grid Nodes in the Differential Altitude Representation.

The two-bit relative difference codes that are used for the application of the corrections are defined as 00, 01 and 10, for a decrease, no change or increase in differential altitude (Δ). For each local surface, a total of eight bytes of storage is necessary to represent the 25 two-byte original elevations, a storage saving of 84%. If the original elevations were stored in the smallest number of possible bits, the savings would still be quite attractive, 68% for ST06 and 71.56% for ST08. The effective number of bits used to represent each elevation in the local surface is 2.56. The local surfaces are stored in two 4-byte values (Figure 5.4).

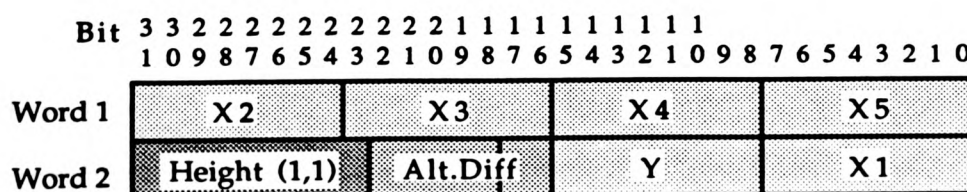


Figure 5.4 - 2-Bit Differential Altitude Storage Representation of 5x5 Local Surface Grid.

Alternatively, three-bits could be used to represent the step or change in differential altitude between grid nodes. In this instance, the correction to neighbouring elevations can be up to $\pm 3 \Delta$. The use of this extended delta modulation code allows the 5x5 grid to be stored as three 4-byte values (Figure 5.5), with a 10-bit Δ value and 10-bit initial elevation.

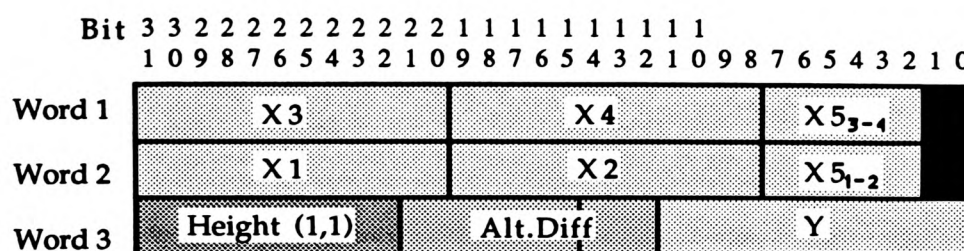


Figure 5.5 - 3-Bit Differential Altitude Storage Representation of 5x5 Local Surface Grid.

The storage saving using this representation is 76% compared to the original two-byte values, or 52% for ST06 and 57.33% for ST08 when considering their compressed minimal storage representations. The effective number of bits used to represent each elevation in the local surface is 3.84. A summary of the storage costs for both methods is presented in Table 5.1.

Model	Bits per Value	% Storage	% of Min Storage
ST06	2.56	16.00	32.00
	3.84	24.00	48.00
ST08	2.56	16.00	28.44
	3.84	24.00	42.67

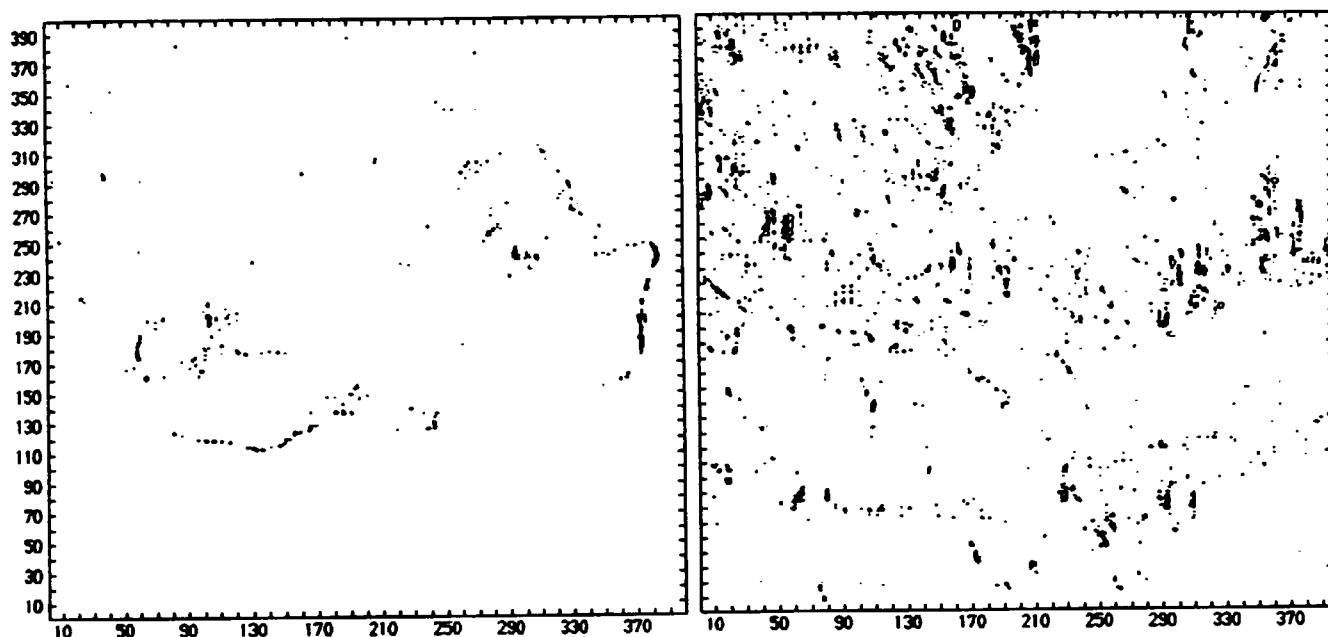
Table 5.1 - Summary of Differential Altitude Storage Statistics for ST06 and ST08.

The elevation errors associated with these surface representations are shown below in Table 5.2, for both two and three bit difference representations of ST06 and ST08.

Model	Abs.Ave. Error	RMSE	Stan. Dev.	% of Points within			Error Range
				±5 m	±10 m	±15 m	
ST06	0.604	1.433	1.55	98.371	99.697	99.929	-30.00 to 38.00
	0.395	0.774	0.87	99.911	99.999	100.000	-10.40 to 8.00
ST08	2.080	3.343	3.94	88.951	97.864	99.415	-36.75 to 41.50
	1.556	2.281	2.76	94.520	99.834	99.999	-13.60 to 17.60

Table 5.2 - Comparison of Differential Altitude Grids with O.S. ST06/08 50m Grids.
(N.B. Top Line = 2 bits, Bottom Line = 3 bits; Errors in metres).

Despite the good overall performance of the methods, the maximum error within the model cannot be constrained or estimated beforehand. Some large errors (ie. $\geq \pm 30$ metres) may therefore occur in regions of variable terrain using the two-bit corrections. The range of elevation differences for grid elements was shown to be ± 56 metres for ST08, which cannot be modelled efficiently in one step. However, the maximum error in the three-bit correction model gives a more satisfactory representation for a small increase in storage. The positions of the largest errors associated with the two-bit differential altitude representation are shown below in Figure 5.6, for both ST06 and ST08.



*Figure 5.6 - Residual Errors in 2-Bit Differential Altitude Representation for ST06 and ST08.
(Errors shown at 10m intervals).*

A comparison of these error maps with the original terrain (in Appendix A) shows that these largest errors occur in regions of variable terrain (ie. coastline, cliffs, hills and mountains, where sharp changes in gradient are predominant). The greater flexibility of the encoded three-bit relative differences allows a much better representation, since instead of a unit increase/decrease or no change in differential altitude between grid nodes, up to ± 3 differences are allowable. This ensures that the modelled points are more likely to resemble their original heights in these areas of steep slopes or changes of gradient. The largest absolute error in the three-bit representation is 17.6 metres for ST08, this being the only value of the 400x400 grid exceeding 15 metres.

As with all data compressed DEMs, applications such as profiling may be performed in one of two ways. The first method uncompresses all the data into its original form of a regular grid

DEM, entailing a fixed 'one-off' overhead of retrieval and construction time for each 20x20 kilometre surface model. This ensures that profile generation time will be the same as for the original 50 metre regular grid DEM (ie. on average, less than 0.3 milliseconds each, for 1000 profiles of between 2 and 28 kilometres). The time taken to uncompress all of the differential altitude grids into the form of the original grid DEM is just over one second of CPU time (approximately 1075 milliseconds), which corresponds to about 6 local patches of 5x5 grid nodes per millisecond, for both two and three bit differential altitudes.

The second approach is more complicated, since the patches that are uncompressed are those surfaces which are needed at the application stage, that is, patches intersected by the profile radial. As discussed earlier, the elevations are considered to represent the heights at the centre of each grid cell, so that duplication along patch boundaries is avoided. However, for the interpolation of points near the boundaries of the local surface, two or four patches may have to be accessed to bilinearly interpolate elevations. The average interpolation time for the 1000 test profiles using this approach is approximately 13.5 milliseconds for both the two and three bit altitude difference models. This represents a profile retrieval time equivalent to 45 times greater than the dense grid DEM, but without the one second overhead of uncompressing the complete model. This overhead corresponds to about 80 profile retrievals, so if it is likely that more profiles than this need to be interpolated, then the complete grid should be reconstructed beforehand, otherwise local surface retrieval should be advocated.

The attenuation losses of the 1000 interpolated profiles from the differential altitude models were calculated and compared to the losses from the original dense 50 metre regular grid DEMs of ST06 and ST08. Table 5.3 shows the radio path loss errors for the two and three bit models for ST06, whilst Table 5.4 records these corresponding errors for ST08.

Freq. (MHz)	ST06 - 2 Bit Differences				ST06 - 3 Bit Differences			
	A.Av. Error	RMSE	No. of Profs ≤ 6dB 12dB		A.Av. Error	RMSE	No. of Profs ≤ 6dB 12dB	
200	0.984	2.069	974	996	0.735	1.745	981	998
400	0.943	2.105	970	996	0.699	1.660	980	999
600	0.906	1.963	971	997	0.688	1.603	977	1000
900	0.928	2.025	966	996	0.676	1.574	975	999
1400	1.033	2.275	963	994	0.748	1.753	972	999
1800	1.091	2.424	959	992	0.788	1.856	972	999

Table 5.3 - ST06 Radio Path Loss Errors for Differential Altitude Grids at 200 to 1800 MHz.

Freq. (MHz)	ST08 - 2 Bit Differences				ST08 - 3 Bit Differences			
	A.A.v. Error	RMSE	No. of Profs ≤ 6dB 12dB		A.A.v. Error	RMSE	No. of Profs ≤ 6dB 12dB	
200	1.706	3.228	939	985	1.365	2.911	959	985
400	1.611	3.137	939	986	1.269	2.532	961	992
600	1.428	2.968	940	986	1.296	2.829	948	984
900	1.391	2.949	941	988	1.198	2.754	952	989
1400	1.434	3.152	949	982	1.241	2.919	956	987
1800	1.513	3.368	944	980	1.304	3.108	952	985

Table 5.4 - ST08 Radio Path Loss Errors for Differential Altitude Grids at 200 to 1800 MHz.

On average, for all frequencies, 96.7% and 94.2% of the two bit relative differential altitude grids for ST06 and ST08 are within 6 dB radio path loss error, whilst the corresponding number for the three bit representations are 97.6% and 95.5% respectively. The overall difference between these average results for the two and three bit representations is quite small (+0.9% and +1.3% improvement for ST06 and ST08 respectively), suggesting that in terms of radio path loss, there are no significant increases in performance to be obtained by choosing the three bit altitude differences. In the worst case, no more than 2% of profiles will be in corresponding path loss error by more than 12 dB for the two bit model, which can be considered negligible due to the errors that are associated with the path loss algorithm for any terrain representation. In terms of the storage costs for these representations (16% and 24% of two byte regular grid elevations), the radio path loss performance is very good.

The encoding of relative altitude differences by an extension of delta modulation to two and three bit representations allows an efficient compression of the original grid DEMs. This extension removes a significant proportion of the 'slope overload' or large elevation errors that can arise in 1-bit systems and also overcomes the problem of granular noise, by allowing flatter segments to be represented by a 'no change' flag. Comparisons with the original two-byte grid representation for both the two and three bit codes produces significant storage savings (84% and 76% respectively). In the worst case (for the two-bit code), only 2.14% of points will be in error by more than 10 metres (for ST08), whilst nearly 89% of the surface representation will be in error by less than 5 metres. The application of these codes to local surface patches (5x5 nodes) allows a compromise between representation and retrieval, since the accuracy of larger or more global surfaces decreases steadily as the area of coverage increases, whilst reconstruction of the 5x5 patches can be accomplished in a reasonable time. Profile interpolation is approximately 45 times slower than for the original grid DEM, but this can be overcome for a 'one-off' grid reconstruction time of one second of CPU time. However, the difference in radio path loss errors between the two models is quite small, suggesting that the

magnitude of the elevation errors is not as significant as having a reduced number of data values, such as with sub-sampled grids. In the worst case, an average of 94.2% of profiles can be retrieved to within a corresponding radio path loss of ± 6 dB.

Some alternative elevation data compression algorithms are considered by Dutton (1983) and Shaffer (1989). Both consider local refinements to approximated terrain in a hierarchical data structure such as a pyramid or quadtree. These methods are similar to the differential altitude grid in some respects, since compact codes are used to modify the estimation of elevations at each level. Both approaches can compress a 16-bit elevation grid into less than 3 bits per elevation (> 80% storage saving). However, prototype models for ST06 and ST08 suggest that the results of both methods are very similar to the differential altitude grid in that they cannot adapt to the variability of rough terrain. In this respect the differential altitude grid performs better, particularly the 3-bit coded difference model.

5.3 Huffman Encoded Grids

A similar approach to surface representation by differential altitude grids is to improve upon its efficiency by the application of a practical compression routine, which maintains the fidelity of the data by identifying the redundancy in grid DEMs. Such compaction techniques are most commonly used in the field of data communications, but the same theory holds for minimising storage capacity. The method proposed predicts neighbouring grid elevations using a simple function, but the errors or corrections are compressed by Huffman coding. This removes the redundancy in the data, since the variable-length codes are assigned according to the variability of the terrain.

The data redundancy of grid DEMs exists because elevations are stored as fixed-length values (eg. 16 bits), and a dense grid of points is necessary to encapsulate all surface features, since no distinction is made between elevations, such as local terrain variability. These problems have primarily arisen from an attempt to model terrain for program and application simplicity, rather than in an effort to represent surfaces efficiently. The ideal approach for storage efficiency is to assign variable-length codes to grid heights, according to some statistical criterion relating elevations. This statistical encoding would take advantage of the probabilities of occurrence of symbols (ie. relationship between elevations), so that short codes can be used to represent frequently occurring symbols, while longer codes are used to represent less-frequently encountered symbols (Held, 1987). Thus the total number of bits constituting the data can be minimised, in a similar manner to that of the Morse Code, whereby common symbols are given short codes, while longer codes are assigned to characters that appear less frequently. Such a code is known as a frequency-dependent code, another example of which is Huffman coding (Huffman, 1952).

For any variable-length coding scheme, the number of different characters or symbols is n . If $p(i)$ is the probability of the i th symbol and the length of a message is $l(i)$, the number of coding digits (bits) assigned to it, then the average message length is

$$l_{ave} = \sum_{i=1}^n p(i) l(i) \quad \dots [5.1]$$

The average information per single symbol is

$$H_{ave} = - \sum_{i=1}^n p(i) \log_2 p(i) \quad \text{bits/symbol} \quad \dots [5.2]$$

This equation represents the mathematical definition of entropy, a term used in information theory to denote the average number of bits required to represent each symbol of a source alphabet (for derivation, see Held, 1987). The efficiency of a code can be calculated as

$$\text{EFFICIENCY} = (H_{ave} / l_{ave}) \times 100\% \quad \dots [5.3]$$

Huffman (1952) developed a procedure for encoding a statistically independent source in such a way as to yield the minimum average code length, or most efficient code. This optimum or minimum-redundancy code, has some basic restrictions imposed on its construction :

- (a) No two symbols consist of identical arrangements of coding digits.
- (b) The symbol codes are constructed such that no additional indication is necessary to specify where a symbol code begins and ends, once the starting point of the sequence is known.
- (c) The length of a given code can never be less than the length of a more probable code. Thus for an optimum code :

$$\begin{aligned} p(1) \geq p(2) \geq p(3) \geq \dots \geq p(n-1) \geq p(n) \quad \text{and} \\ l(1) \leq l(2) \leq l(3) \leq \dots \leq l(n-1) = l(n) \end{aligned} \quad \dots [5.4]$$

The code that will be considered here is a binary (two-state) symbol code. Thus to use Huffman's algorithm, a binary coding tree needs to be constructed as follows :-

- (i) Arrange the source probabilities of the symbols (nodes) in descending order.
- (ii) Commencing with the symbols with the two lowest probabilities p_1 and p_2 , construct a new node of which these two probabilities are branches, the new node being labelled with the arithmetic sum of these two probabilities.
- (iii) Repeat the process using the new node instead of the original two, until only one node is left, with a label probability of 1.00.
- (iv) Label each upper branch '0' and the lower branch '1', or vice versa.
- (v) The code for each of the original symbols is then found by proceeding from the root of the tree to the required leaf, noting the branch label of each node traversed.

For example, consider the following seven symbol code, with the probabilities 0.35, 0.27, 0.12, 0.09, 0.09, 0.05 and 0.03. The Huffman codes for these probabilities are calculated below as:

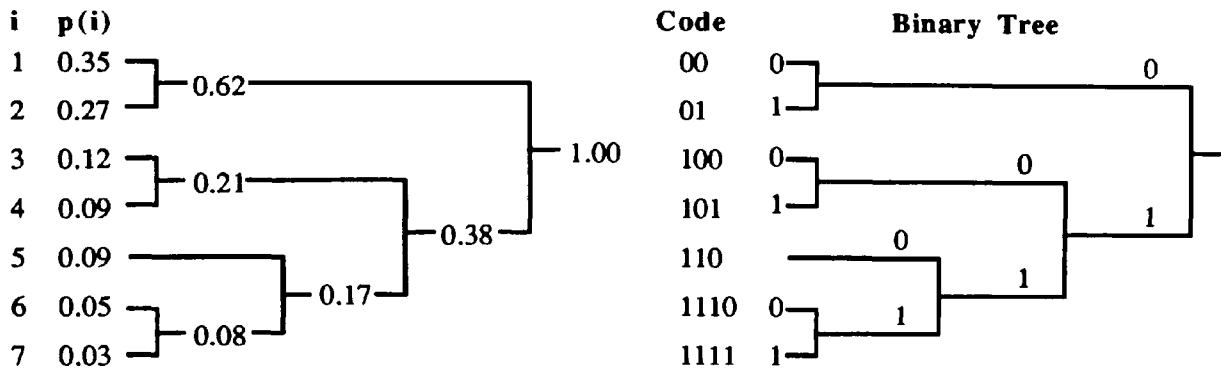


Figure 5.7 - Illustration of the Formation of Huffman Codes to Probability Distributions.

From Equations [5.1] and [5.2], the average code length is calculated as 2.46 bits/symbol and the entropy as 2.4005 bits/symbol. Hence, from [5.3] the efficiency of this code is 97.58%, compared to 80.02% for a fixed-length code of three bits to represent the seven symbols.

Once the codes have been defined for all the symbols (eg. elevation corrections), the data set can be written to a file as a series of packed codewords in strings of bits. At the decoding stage of each symbol, each bit is read in turn until a valid codeword is determined. A list of valid codewords and their corresponding decoded values is therefore maintained to check whether the string of bits (ie. current codeword) exists. If it does, then its corresponding correction value, for example, can be returned as the decoded symbol. The next bit will therefore be the start bit of the next codeword. Alternatively, if there is no matching code, the current codeword is extended one bit at a time, until a match is found.

To facilitate the matching procedure, it was found that the search time can be reduced significantly by only checking those codewords in the look-up table (ie. the list of codewords and corresponding corrections), that have the same number of bits as the current codeword. This is accomplished with an index which indicates the first position in the look-up table for codewords of the same bit-length. As each bit of the current codeword is read, the occurrence of codes with the same bit-length is recorded from the index. If this is non-zero, the codewords in the look-up table can be searched for a match, from the position indicated by the pointer index. The benefits of the index into the look-up table (ie. reduced search time) will become greater as the number of codewords increases.

A grid of elevations is not the ideal choice for Huffman encoding, since the frequency or probability distribution of heights will be fairly uniform, resulting in the codes being of similar bit-length. The number of codes will also be equal to the number of different elevations, which

could possibly be hundreds. Huffman coding is at its most efficient when applied to a distribution of values that is skewed. Values that have a high probability are then given a short code, whilst less frequent occurrences are given longer codes. The problem of application to a regular grid can be overcome by considering the height differences between points, rather than the elevations themselves. In this way, a distribution of elevation corrections can be formed, which actually takes into account the variability of the terrain.

The simplest method of encoding the elevations for Huffman coding is to use a differential altitude representation, such as in Section 5.2. The differences in altitude from neighbouring grid nodes (along the rows) was shown to be -42 metres to +33 metres for ST06, and ± 56 metres for ST08. The distribution of these differences is then suitable for Huffman coding, since the most probable corrections are in the middle of these distributions and the least frequent corrections are at the extremes.

The most common approach to forming the elevation differences is to record the corrections needed to represent neighbouring elevations, usually in a continuous sequence through the grid or from a reference height at the start of each row or column. Consider the 16x16 sub-grid of the O.S. ST08 grid below (Table 5.5), sampled at 50 metre intervals and representing a 750x750 metre region. These heights would normally be stored as 16 bit values, but since the elevation range of this surface is 327 to 470 metres, each individual height could be stored in a maximum of 9 bits (range 0-511 metres). Alternatively, if the nodes were stored as the elevations minus the minimum height (0 to 143 metres), the data could be compressed into 8 bits (range 0-255). Further compression is possible by assigning seven bit codes to each elevation occurrence within this new relief range, since there are only 110 distinct elevation classes. However, with this and other representations, the minimum amount of storage is variable for different types of terrain, so a common standard storage unit size cannot be adopted without some redundancy.

338	333	332	327	337	343	347	352	358	369	375	381	393	405	413	420
352	349	345	339	350	351	361	359	360	365	373	378	386	397	406	413
368	364	359	354	360	364	369	367	364	369	372	378	385	393	401	408
382	375	368	363	369	372	375	377	373	371	375	379	384	390	397	402
394	386	378	373	375	378	381	384	385	380	379	382	385	390	394	399
407	396	388	381	381	384	386	389	391	390	384	390	390	390	393	395
418	409	399	392	390	390	391	392	392	391	388	390	390	391	391	392
426	420	409	403	399	398	398	396	395	393	390	390	390	391	390	389
435	428	422	414	410	405	402	399	398	395	391	389	387	385	386	385
448	443	433	425	419	410	407	402	401	398	393	388	383	380	378	379
454	453	447	441	432	426	420	409	405	401	395	388	382	377	370	372
457	457	454	450	444	438	430	419	408	404	398	390	384	378	370	366
459	460	459	458	451	448	442	431	417	408	402	393	387	381	372	366
459	461	463	462	459	456	452	442	427	414	405	396	388	382	374	367
458	462	465	468	465	461	457	449	438	421	409	397	389	382	375	368
454	461	467	470	470	464	461	454	445	430	413	400	389	383	375	369

Table 5.5 - 16x16 Sub-Grid of ST08 Sampled at 50m, with 15x15 Model Region Highlighted

The elevation differences or corrections that are applied to the source elevations are calculated from the first value in each row and are shown below in Table 5.6 for the data in Table 5.5.

338	-5	-1	-5	10	6	4	5	6	11	6	6	12	12	8	7
352	-3	-4	-6	11	1	10	-2	1	5	8	5	8	11	9	7
368	-4	-5	-5	6	4	5	-2	-3	5	3	6	7	8	8	7
382	-7	-7	-5	6	3	3	2	-4	-2	4	4	5	6	7	5
394	-8	-8	-5	2	3	3	3	1	-5	-1	3	3	5	4	5
407	-11	-8	-7	0	3	2	3	2	-1	-6	6	0	0	3	2
418	-9	-10	-7	-2	0	1	1	0	-1	-3	2	0	1	0	1
426	-6	-11	-6	-4	-1	0	-2	-1	-2	-3	0	0	1	-1	-1
435	-7	-6	-8	-4	-5	-3	-3	-1	-3	-4	-2	-2	-2	1	-1
448	-5	-10	-8	-6	-9	-3	-5	-1	-3	-5	-5	-5	-3	-2	1
454	-1	-6	-6	-9	-6	-6	-11	-4	-4	-6	-7	-6	-5	-7	2
457	0	-3	-4	-6	-6	-8	-11	-11	-4	-6	-8	-6	-6	-8	-4
459	1	-1	-1	-7	-3	-6	-11	-14	-9	-6	-9	-6	-6	-9	-6
459	2	2	-1	-3	-3	-4	-10	-15	-13	-9	-9	-8	-6	-8	-7
458	4	3	3	-3	-4	-4	-8	-11	-17	-12	-12	-8	-7	-7	-7
454	461	467	470	470	464	461	454	445	430	413	400	389	383	375	369

Table 5.6 - 15x15 Altitude Corrections Calculated from the Differences along Grid Rows.

The range of altitude differences or corrections for this surface is -17 to + 12 metres (30 values), which can be represented in a maximum of 5 bits of storage per correction (ie. range 0-31). This represents a significant decrease in the number of elevation classes compared to the original 110 different elevation values of the original surface. The redundancy in this representation is quite small, since the range of corrections approaches the maximum possible representations of the fixed-length code. However, the redundancy that does exist can be reduced significantly by the application of Huffman coding, to produce optimal variable length codes for the altitude differences. The distribution of differences is more suited to this form of coding, since the probability distribution of occurrences is skewed. The formation of the codes by grouping the probabilities is illustrated below in Figure 5.8 for the relative altitude differences of Table 5.6. This is accomplished in a similar manner to the example in Figure 5.7, whereby each codeword is traced from the root node, recording the bit value at each branch as either a 0 or 1 (top or bottom branch). The frequency of correction occurrences are shown here for an easier overview, rather than the probabilities themselves.

It can be seen that the more probable altitude differences are given shorter codes than less common corrections, thus satisfying the Huffman condition of equation [5.4]. For this probability distribution, the entropy is calculated from equation [5.2] as 4.459 bits per altitude difference, whilst the average codeword length is 4.489, giving a code efficiency of 99.34%. At different stages in the formation of the binary tree of Figure 5.8, the choice of the two lowest probabilities is far from unique. This may result in numerous, equally valid forms of the Huffman codewords, with some being of different length in each representation. However, in each case the average code length will be the same, since the algorithm is guaranteed to

produce an optimal (minimum redundancy) code.

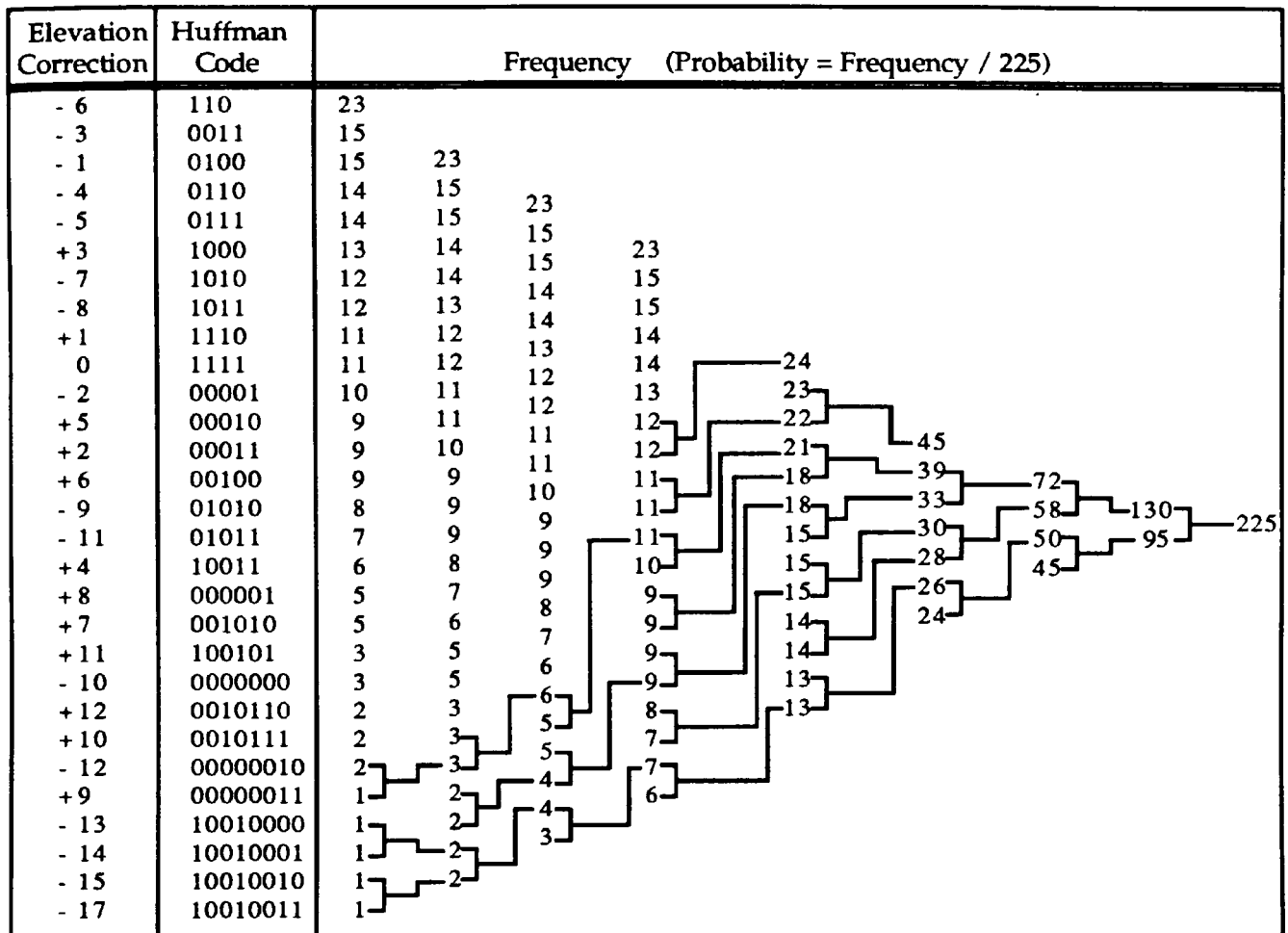


Figure 5.8 - Formation of Huffman Codes for the Frequency Distribution of Altitude Differences.

Since this coding scheme is optimal, further improvements to data compression can only be achieved by creating a more skewed distribution of altitude corrections, to which the Huffman coding can be applied. A number of alternative methods to the differencing approach were examined to get a better distribution of values. These methods predict elevations based on local terrain variation, rather than using a simple differencing approach, and the error corrections are used in the same way as before. The techniques investigated included slope (or gradient) analysis and simple polynomial modelling. The method that was found to give the best performance was an extrapolation of the three neighbouring elevations of a grid node. The method is based on the assumption that the linearly interpolated mid-point between a pair of diagonal grid nodes will be equivalent in elevation to the linearly interpolated mid-point of the other intersecting diagonal forming the grid square, as in Figure 5.9.

The difference in elevation between this height (d) and the actual elevation is used as the correction for Huffman coding. This process is repeated along the row, for all rows in the grid.

The method assumes that the first row and column of each grid DEM are known, rather than the first row or column of the altitude difference method. The range of elevation corrections compared to the standard altitude difference method above, will be reduced significantly, with a better probability distribution more suited for Huffman encoding.

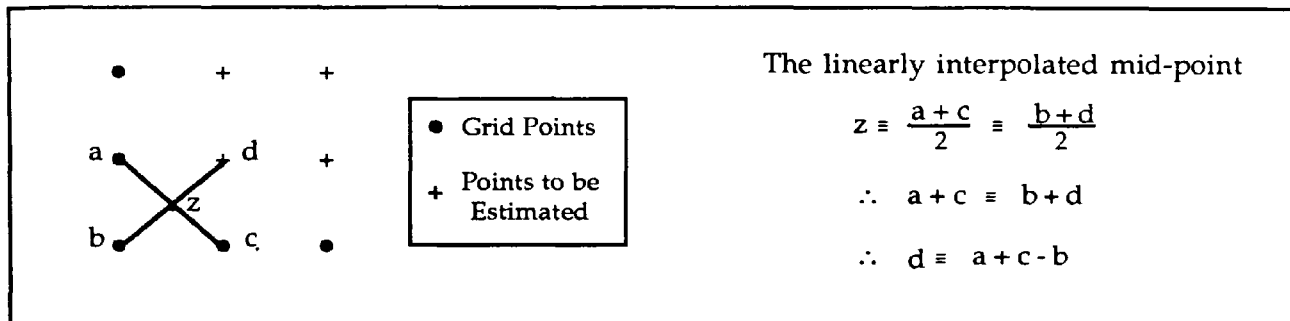


Figure 5.9 - Prediction Algorithm used for Estimating Grid Elevations.

This method was applied to the test sample data set of Table 5.5, to produce a grid of elevation corrections (Table 5.7). It is noticeable that the magnitude of these corrections is smaller than for the relative altitude differences of Table 5.6. The range of corrections is smaller (-6 to +7 metres), thus requiring a maximum of only four bits of storage for this representation.

338	- 2	3	1	- 1	5	- 6	7	5	6	- 2	1	4	1	- 1	0
352	1	1	- 1	5	- 3	5	0	4	0	5	- 1	1	3	1	0
368	3	2	0	0	1	2	- 4	1	7	- 1	2	2	2	1	2
382	1	1	0	4	0	0	- 1	- 5	3	5	1	2	1	3	0
394	3	0	2	2	0	1	0	- 1	- 4	5	- 3	3	5	1	3
407	- 2	2	0	2	3	1	2	2	0	- 3	4	0	- 1	3	1
418	- 3	1	- 1	2	1	1	3	1	1	0	2	0	0	1	2
426	1	- 5	2	0	4	3	1	0	1	1	2	2	3	- 2	0
435	- 2	4	0	2	4	0	2	0	0	1	3	3	1	3	- 2
448	- 4	- 4	- 2	3	- 3	3	6	3	1	1	2	1	2	5	- 1
454	- 1	- 3	- 2	- 3	0	2	0	7	0	0	1	0	1	1	6
457	- 1	- 2	- 3	1	- 3	- 2	0	3	5	0	1	0	0	1	2
459	- 1	- 3	0	- 4	0	- 2	- 1	1	4	3	0	2	0	- 1	1
459	- 2	- 1	- 4	0	1	0	- 2	- 4	4	3	3	0	1	- 1	0
458	- 3	- 3	0	- 3	2	- 1	- 1	- 2	- 2	5	1	3	- 1	1	- 1
454	461	467	470	470	464	461	454	445	430	413	400	389	383	375	369

Table 5.7 - 15x15 Altitude Corrections Calculated from the Prediction Algorithm

The corresponding Huffman codes for these corrections (Figure 5.10) illustrate the suitability of a more skewed probability distribution for coding. As the method of prediction becomes more accurate, the number of codewords (ie. the range of corrections) becomes less, and the probabilities of those codewords in the model will become more skewed. For this probability distribution, the entropy is 3.270 bits per correction, whilst the average codeword length is 3.324 bits. This corresponds to a code efficiency of 98.38 %. The relative saving in average code length over the standard altitude difference method is 1.165 bits per elevation, or a reduction of

25%. The probability distributions for both of these altitude difference methods applied to the test data set are illustrated below in Figure 5.11.

Elevation Correction	Huffman Code	Frequency (Probability = Frequency / 225)									
+ 1	10	45	45	45	45						
+ 0	000	44	44	44	45						
+ 2	011	27	27	27	44	45					
+ 3	110	24	24	24	27	44					
- 1	111	21	21	21	24	28	55				
- 2	0100	15	15	15	21	27	45	90			
- 3	0101	13	13	13	15	24	45	80			
+ 5	00100	11	11	11	13	21	44	55	135		
+ 4	00101	9	9	9	11	20	36	90	225		
- 4	00111	7	7	7	9	16					
+ 7	001101	3	3	3	6	7					
+ 6	0011000	3	3	3	3	7					
- 5	00110010	2	2	2	2						
- 6	00110011	1	1	1	1						

Figure 5.10 - Formation of Huffman Codes for the Frequency Distribution of Predicted Corrections.

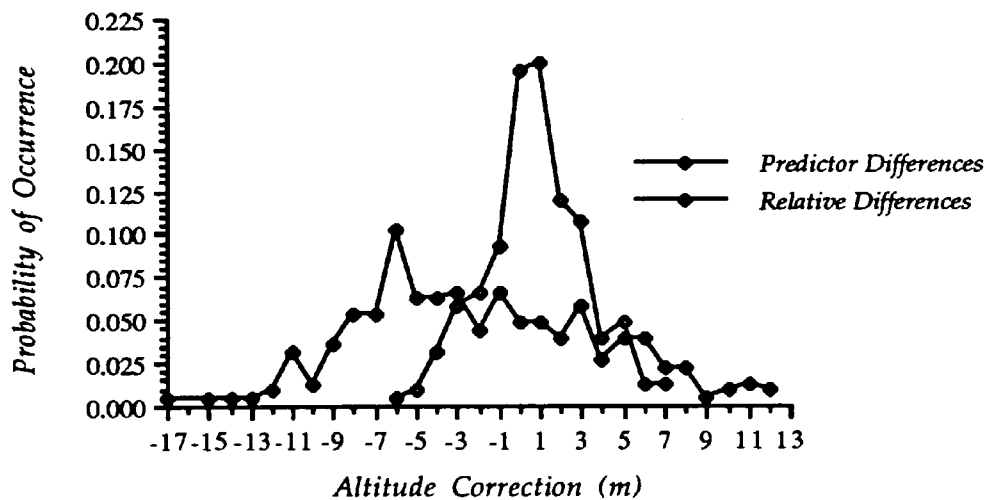


Figure 5.11 - Probability Distributions for the Differential Altitude Grids of Tables 5.6 & 5.7.

The shape of these probability distributions highlights their respective suitability for Huffman coding. The use of short codes for very probable occurrences, and longer codes for less common occurrences, illustrate clearly that the probability distribution for the differences of the predicted elevations is more suited to Huffman coding. A summary of the storage costs of the methods described is given below in Table 5.8 for this test data set.

The compression ratios of these techniques are very promising, especially since the original terrain can be reconstructed without incurring any error. From this table, it can be seen that the benefits to be gained from Huffman coding the altitude differences is quite small, when

compared to a fixed length code (ie. improvement of less than 5 % of original two-byte storage). However, this is primarily due to the range of altitude differences being quite small, resulting in the ability to encode them as short fixed length codes. For larger areas of terrain, this range of altitude differences would increase significantly, resulting in longer fixed codes, whilst the fluctuation in average bit length of the Huffman codes would be minimal.

Elevation Representation	Values	Ave.Min. No.of Bits	% Storage of Min. Bits (9)	% Storage of Max. Bits (16)
Maximum Elevation	470	9.000	100.00	56.25
Elevation Range (327 to 470m)	144	8.000	88.89	50.00
Distinct Elevation Values	110	7.000	77.78	43.75
Relative Altitude Diffs (-17 to 12m)	29	5.000	55.56	31.25
Huffman Coded Rel. Alt. Diffs.	29	4.489	49.88	28.06
Predicted Altitude Diffs (-6 to 7m)	14	4.000	44.44	25.00
Huffman Coded Pred. Alt. Diffs	14	3.324	36.94	20.78

Table 5.8 - Storage Performances for the 15x15 Test Grid Compression Techniques .

To illustrate this fact, the altitude difference methods were applied to the 401x401 grid DEMs of ST06 and ST08 and the resulting elevation corrections were Huffman encoded. For the predicted elevations, it is assumed that the elevations of the first row and column are known, since to extrapolate elevations in the second row and column, the three values of each enclosed grid cell are required. However, the first row and column could themselves be Huffman coded by the relative altitude difference method and decoded using the same codes as for the prediction algorithm. In this instance, only the first element of the grid DEM needs to be stored in its original elevation form, for both altitude difference algorithms. The storage costs for both these methods applied to ST06 and ST08 are shown below in Tables 5.9 and 5.10.

ST06 Elevation Representation	Values	Ave.Min. No.of Bits	% Storage of Min. Bits (8)	% Storage of Max. Bits (16)
Elevation Range (0 to 135m)	136	8.000	100.00	50.00
Relative Altitude Diffs (-32 to 42m)	66	7.000	87.50	43.75
Huffman Coded Rel. Alt. Diffs.	66	2.505	31.31	15.65
Predicted Altitude Diffs (-40 to 41m)	52	6.000	75.00	37.50
Huffman Coded Pred. Alt. Diffs	52	1.968	24.60	12.30

Table 5.9 - Storage Comparisons of Altitude Difference Grids with Original Grid DEM for ST06.

ST08 Elevation Representation	Values	Ave.Min. No.of Bits	% Storage of Min. Bits (9)	% Storage of Max. Bits (16)
Elevation Range (19 to 470m)	452	9.000	100.00	56.25
Relative Altitude Diffs (-56 to 52m)	95	7.000	77.78	43.75
Huffman Coded Rel. Alt. Diffs.	95	4.634	51.49	28.97
Predicted Altitude Diffs (-39 to 49m)	59	6.000	66.67	37.50
Huffman Coded Pred. Alt. Diffs	59	3.238	35.97	20.23

Table 5.10 - Storage Comparisons of Altitude Difference Grids with Original Grid DEM for ST08.

The saving in storage using the Huffman coded differences is considerably better than that achievable for fixed-length codes. As the model region of coverage increases, different terrain types are likely to be encountered, which will increase the number of corrections or codewords needed. For fixed-length codes, as the number of codewords increases by a factor of two, the length of the code needed to represent all possibilities will increase by one bit, which can become significant. No distinction is made between correction values and their relative importance or probability. However, with Huffman codes the average code bit length remains at a more uniform level, since the corrections are ranked according to probability. This means that the introduction of more codewords does not have any significant effect, since if they are rare occurrences they will be assigned a longer code which will not significantly affect the average bit length of the code. This is illustrated in Table 5.11 for the Huffman encoded predicted altitude differences for ST06.

Rare correction occurrences are assigned codes of up to 18 bits in length, whilst the most common correction is assigned a one-bit code, since its probability is greater than 50%. The significance of the longer codes for average code length is thus quite small. Table 5.11 shows that 102,277 grid elevations (or 63.92% of the data set) need no corrections applied to the predicted elevations. The code to signify this fact is only one bit in length, the minimum possible in a surface that is not single-valued. The reason why this number of occurrences is so high is primarily due to the fact that 56,513 grid nodes (35.32%) represent sea level elevations. By representing these values by a one-bit Huffman code, a significant amount of the redundancy of the original two-byte elevations can be removed.

The savings in storage achieved by Huffman coding the altitude differences are 84.35% for ST06 and 71.03% for ST08 using the relative differences, whilst corresponding savings of 87.7% and 79.77% are attained using the differences of the prediction algorithm. These storage figures are based on comparisons with the original two-byte elevations. In all cases there is a small additional overhead of storing the codes and their corresponding corrections (differences), together with an optional look-up table. However, Huffman coding of altitude differences

Elevation Correction	Number of Occurrences	Number of Bits	Huffman Code
-40	1	18	010011011000101110
-33	1	18	010011011000101111
-29	1	18	010011110001111000
-26	4	16	0100111100011111
-22	2	17	01001111000111101
-21	1	18	010011110001111001
-20	7	15	010011110001101
-19	3	16	0100111100010000
-18	3	16	0100111100010001
-17	7	15	010011110001110
-16	6	15	010011110001001
-15	6	15	010011110001100
-14	10	14	01001101111101
-13	12	14	01001111000101
-12	34	12	010011011001
-11	32	13	0100111101011
-10	59	12	010011110100
-9	75	11	01001101110
-8	109	11	01001111001
-7	194	10	0100111011
-6	286	9	010011010
-5	473	9	01001111
-4	965	7	0100001
-3	2067	6	010010
-2	5189	5	01011
-1	19253	3	011
0	102277	1	1
+1	19427	2	00
+2	5146	5	01010
+3	2035	6	010001
+4	951	7	0100000
+5	523	8	01001100
+6	301	9	010011100
+7	161	10	0100111010
+8	128	11	01001111011
+9	74	11	01001101101
+10	45	12	010011110000
+11	39	12	010011011110
+12	29	13	0100111101010
+13	23	13	0100110111111
+14	9	14	01001101100011
+15	8	14	01001101100000
+16	4	15	010011011000010
+17	9	14	01001101111100
+18	2	16	0100110110000110
+19	2	16	0100110110000111
+20	1	17	01001101100010000
+21	1	17	01001101100010001
+23	1	17	01001101100010010
+25	1	17	01001101100010011
+26	2	16	0100110110001010
+41	1	17	01001101100010110

Table 5.11 - Huffman Encoded Predicted Altitude Differences for ST06.
(Elevation Differences of -40 to +31 metres represented by Codewords of 1 to 18 bits).

shows that considerable savings in storage can be attained by removing the redundancy within the grid terrain representation. The predicted elevations give the best estimate for the differences, resulting in grid values being represented by an average of 1.968 bits for ST06 (entropy of 1.881 bits per elevation and code efficiency of 95.58%) and 3.238 bits for ST08 (entropy of 3.183 bits per elevation and code efficiency of 98.32%).

A disadvantage with Huffman encoding is the complexity of the decompression process. Since most computers are word rather than bit oriented, the length variability of the Huffman code for different characters is usually considered a drawback. By the same reasoning, a fixed length code of say, five or six bits, will also be difficult to decode efficiently without bit operations. The basic method for interpreting each Huffman codeword is to read each bit in sequence and to search for a valid codeword in the look-up table. To uncompress the 400x400 grid nodes of ST06 and ST08, the number of bits or individual check operations are 314,894 and 518,006 respectively. This is very time consuming, resulting in grid reconstruction times of 3.27 and 5.09 seconds of CPU allocation respectively. This corresponds to approximately 100 bits per millisecond, or 45 and 30 elevations per millisecond for ST06 and ST08 respectively.

The index and look-up table used to decode the Huffman codes of Table 5.11 (which accounts for less than 0.2% of original storage), are illustrated below in Table 5.12. Only the frequency count for codewords of similar bit-length and the codewords themselves with their corresponding corrections need to be stored. The pointer values can be formulated from this frequency count during retrieval.

However, it can be seen from the cumulative percentage of points for the codes, that the vast majority of altitude corrections (> 99%) are represented by a minority of the codes (ie. between 11 and 13 of the 52 codewords). In this instance, an alternative approach to storing the corrections might be to use a code constrained to a maximum length, or an optimal length-limited Huffman code. This combines the use of a variable length code with a fixed length code. For example, the variable length Huffman code could be used to represent 95% of the most common codeword corrections (which in this example would be the first seven codes of up to six bits in length, representing 97.121% of the corrections). Another codeword of the same maximum bit length could then be constructed to signify that the actual correction value is encoded in the following six bits (for up to 64 possibilities). Hence all the remaining codes can be represented by strings of 12 bits. The complete coding scheme would allow all of the most probable codewords to be one to six bits in length, whilst the least probable codewords would be 12 bits. The advantage of such a coding scheme is that the search time for lengthier codes can be reduced significantly, since they can be directly retrieved after the first six bits have been read. For ST08, the one to six bit codes represent the first 11 codewords or 95.108% of all corrections. Preliminary results have shown that by specifying maximum length codewords,

grid retrieval time can be improved by over 15%.

No.of Bits	Freq.	Pointer	Cumulative % of Points		Elevation Correction	Huffman Code
1	1	1	63.923		0	1
2	1	2	76.065		+ 1	00
3	1	3	88.098		- 1	011
4	0	4	88.098	:	- 2	01011
5	2	4	94.558	:	+ 2	01010
6	2	6	97.121	:	- 3	010010
7	2	8	98.319	:	+ 3	010001
8	1	10	98.646	:	- 4	0100000
9	3	11	99.308	:	+ 4	0100001
10	2	14	99.530	:	+ 5	01001100
11	4	16	99.771	:	- 5	010011111
12	4	20	99.882	:	+ 6	010011100
13	3	24	99.934	:	- 6	010011010
14	5	27	99.964	:	- 7	0100111011
15	5	32	99.983	:	+ 7	0100111010
16	6	37	99.993	:	+ 8	01001111011
17	6	43	99.998	:	- 8	01001111001
18	4	49	100.000	:	- 9	01001101110
				:	+ 9	01001101101
				:	- 10	010011110100
				:	+ 10	010011110000
				:	+ 11	010011011110
				:	- 12	010011011001
				:	- 11	0100111101011
				:	+ 12	0100111101010
				:	+ 13	0100110111111
				:	- 13	01001111000101
				:	- 14	01001101111101
				:	+ 14	01001101100011
				:	+ 17	01001101111100
				:	+ 15	01001101100000
				:	- 17	010011110001110
				:	- 20	010011110001101
				:	- 15	010011110001100
				:	- 16	010011110001001
				:	+ 16	010011011000010
				:	- 26	0100111100011111
				:	- 18	0100111100010001
				:	- 19	0100111100010000
				:	+ 18	0100110110000110
				:	+ 19	0100110110000111
				:	+ 26	0100110110001010
				:	- 22	01001111000111101
				:	+ 20	01001101100010000
				:	+ 21	01001101100010001
				:	+ 23	01001101100010010
				:	+ 25	01001101100010011
				:	+ 41	01001101100010110
				:	- 21	010011110001111001
				:	- 29	010011110001111000
				:	- 33	010011011000101111
				:	- 40	010011011000101110

Table 5.12 - Index & Look-Up Table for ST06 Huffman Encoded Predicted Altitude Differences.

This method has shown that significant storage reductions of regular grid DEMs can be attained by combining predictive coding theory with error-free Huffman data compression. Storage savings of up to 88% can be achieved over fixed two-byte regular grid DEMs, with no elevation or radio path loss error. This is accomplished at a 'one-off' cost of between three and five seconds of CPU time for grid reconstruction, depending on the average code length. Predictive Huffman coding is adaptive to terrain variability, since the elevations of uniform terrain will produce a clustering of corrections that may be represented by short codewords, whilst sharp changes in terrain elevation that cannot be predicted accurately will be represented by longer codewords. Since the coding of elevation differences or corrections using Huffman's algorithm is optimal, further improvements in grid compression can only be achieved with the use of a better prediction algorithm, or the inclusion of constrained error tolerances.

5.4 Error Tolerant Huffman Grids

The use of error-free Huffman encoded DEMs has shown that storage savings of between 79% and 88% were achievable for ST08 and ST06, respectively. However, further compression can be achieved with the inclusion of constrained maximum errors by quantising or banding elevation classes together. This can be accomplished in a number of ways, for example, rounding up or down elevations to even or odd values can reduce the range of distinct elevation classes or sets by up to 50%. By representing these grouped elevations with set numbers, prediction techniques will produce a closer clustering and hence better distribution of differences than for elevation differences. This ensures a more efficient Huffman coding, whilst constraining the maximum error to a specified tolerance.

Terrain models were constructed using this method for constrained error tolerances, in elevation increments of 0.5 metres. An illustration of the bounding of the first four elevation sets is shown below in Table 5.13, together with the heights that correspond to each set. These representative heights (ie. the set mid-point) replace the elevations for each value of that set in the new terrain model. The number of elevations banded together in each set is equal to $(\text{TOLERANCE} \times 2 + 1)$ values, where TOLERANCE is the maximum constrained error. This value also corresponds to the increment in representative height between neighbouring set values.

Maximum Error	Grouped Elevation Sets and Representative Heights			
	0	1	2	3
0.0 m	0	1 (1.0)	2 (2.0)	3 (3.0)
0.5 m	0	1,2 (1.5)	3,4 (3.5)	5,6 (5.5)
1.0 m	0	1,2,3 (2.0)	4,5,6 (5.0)	7,8,9 (8.0)
1.5 m	0	1,2,3,4 (2.5)	5,6,7,8 (6.5)	9,10,11,12 (10.5)
2.0 m	0	1,2,3,4,5 (3.0)	6,7,8,9,10 (8.0)	11,12,13,14,15 (13.0)
2.5 m	0	1,2,3,4,5,6 (3.5)	7,8,9,10,11,12 (9.5)	13,14,15,16,17,18 (15.5)

Table 5.13 - Illustration of Elevation Banding for the First Four Sets.

To illustrate this more clearly, consider the case of all grid nodes at a height, for example, of six metres. For the model with no allowable error (0.0 m), this is clearly represented by a set value of 6 corresponding to an elevation of 6 metres. However, in the constrained error models, it can be seen from Table 5.13 above, that this value would be represented by heights of 5.5 metres (set 3 at a tolerance of 0.5m), 5.0 metres (set 2 at 1.0m), 6.5 metres (set 2 at 1.5m), 8.0 metres (set 2 at 2.0m) and 3.5 metres (set 1 at 2.5m).

The storage costs of quantising the original grid DEMs are shown below in Tables 5.14 and 5.15, for ST06 and ST08. The first half of each table illustrates the cost of representing the quantised elevations by set number using a standard fixed length code, whilst the second half illustrates the further savings obtainable by Huffman-encoding the differences of set prediction values, using the method described in Section 5.3. By tabulating the grouped elevations as fixed and variable length codes, the storage savings of Huffman-encoding the grids can be fully appreciated.

Max. Abs. Error	Height Sets	Min. Bits	% of Min. Bits (8)	% of Max Bits (16)	Huffman Bits	% of Min. Bits (8)	% of Max Bits (16)
0.0	136	8	100.00	50.00	1.968	24.60	12.30
0.5	69	7	87.50	43.75	1.570	19.63	9.81
1.0	46	6	75.00	37.50	1.426	17.83	8.91
1.5	35	6	75.00	37.50	1.356	16.95	8.48
2.0	28	5	62.50	31.25	1.311	16.39	8.19
2.5	24	5	62.50	31.25	1.276	15.95	7.98

Table 5.14 - Storage Costs of Banded Grid DEMs for Fixed and Variable Length Codes of ST06.

Max. Abs. Error	Height Sets	Min. Bits	% of Min. Bits (9)	% of Max Bits (16)	Huffman Bits	% of Min. Bits (9)	% of Max Bits (16)
0.0	452	9	100.00	56.25	3.238	35.98	20.24
0.5	227	8	88.89	50.00	2.450	27.22	15.31
1.0	152	8	88.89	50.00	2.036	22.62	12.73
1.5	114	7	77.78	43.75	1.838	20.42	11.49
2.0	92	7	77.78	43.75	1.725	19.17	10.78
2.5	77	7	77.78	43.75	1.655	18.39	10.34

Table 5.15 - Storage Costs of Banded Grid DEMs for Fixed and Variable Length Codes of ST08.

These tables show that whilst quantising grid values may not produce significant storage savings for fixed length codes, predictive coding can further compress the Huffman-encoded error-free grid by up to 50% (ST08), for a maximum absolute tolerance of 2.5 metres (or 35% for

ST06). However, as a percentage of original grid storage, these compression ratios may not seem too significant an improvement for the inclusion of elevation errors. For ST06, the error-free model (12.3% of original storage) can be reduced to 7.98% of storage for a grid with a maximum elevation error of 2.5 metres. The corresponding storage costs for ST08 are 10.34% at 2.5 metres (compared to 20.24% for the error-free grid).

Grid compression using quantised elevations, together with predictive Huffman coding will therefore allow storage savings approaching 90% or greater for a maximum constrained error of 2.5 metres. Whilst this is a significant improvement on other grid-based methods, the effect of elevation errors were analysed further in terms of overall model errors and radio path loss errors. For some applications, the reduction in storage overheads may need to be compared to the introduction of elevation errors for further consideration as a significant improvement to the error-free Huffman-encoded grid. Table 5.16 below, indicates the elevation errors that are created by quantising the grid DEMs for ST06 and ST08.

Max. Abs. Error	ST06			ST08		
	Abs. Ave. Error	RMSE	Stan. Dev.	Abs. Ave. Error	RMSE	Stan. Dev.
0.5	0.323	0.402	0.505	0.500	0.500	0.697
1.0	0.432	0.657	0.789	0.666	0.816	1.053
1.5	0.648	0.901	1.104	0.999	1.116	1.488
2.0	0.809	1.171	1.362	1.226	1.436	1.829
2.5	0.970	1.374	1.688	1.497	1.705	2.257

Table 5.16 - Comparison of Quantised Grids with O.S. ST06/08 50m Grids (Errors in metres).

In general, the elevation errors are quite small, since they are constrained to maximum limits. However, in certain instances every point in the model will be in error, since the average or mid-point value of the quantised set is at a non-integer value (ie. in the models constrained to tolerances of 0.5, 1.5 and 2.5 metres). The RMS errors for all of these models are less than the corresponding errors associated with the 100 metre regular sub-sampled grids and the 7.5 metre variable density grids. The errors are generally less than for other grid techniques, with the advantage of additional storage savings. The effect that the errors have on radio path performance are illustrated below in Tables 5.17 and 5.18 for ST06 and ST08.

When averaged over all the examined frequencies in the range 200 to 1800 MHz, all of the representations will produce 95% of interpolated profiles within a corresponding radio path loss error of 6 dB, for both ST06 and ST08. For ST06, the performance of the method (in terms of the percentage of profiles within 6 dB), decreases with frequency for the latter three models (1.5 to 2.5 metres), suggesting that some small elevation errors can be critical.

Max. Err.	Frequency = 200 MHz				Frequency = 400 MHz				Frequency = 600 MHz			
	A.A.v. Error	RMSE	No. of Profs ≤ 6dB 12dB		A.A.v. Error	RMSE	No. of Profs ≤ 6dB 12dB		A.A.v. Error	RMSE	No. of Profs ≤ 6dB 12dB	
0.5 m	0.522	1.267	991	999	0.425	1.142	991	999	0.425	1.161	988	1000
1.0 m	0.800	1.769	984	997	0.600	1.353	986	1000	0.601	1.352	988	1000
1.5 m	1.057	1.916	979	998	0.900	1.843	975	998	0.837	1.698	974	1000
2.0 m	1.376	2.532	969	992	1.206	2.284	959	996	1.145	2.152	967	998
2.5 m	1.490	2.525	966	994	1.209	2.241	966	997	1.140	2.121	967	997

Table 5.17a - ST06 Radio Path Loss Errors for Huffman Predicted Differences at 200 to 600MHz.

Max. Err.	Frequency = 900 MHz				Frequency = 1400 MHz				Frequency = 1800 MHz			
	A.A.v. Error	RMSE	No. of Profs ≤ 6dB 12dB		A.A.v. Error	RMSE	No. of Profs ≤ 6dB 12dB		A.A.v. Error	RMSE	No. of Profs ≤ 6dB 12dB	
0.5 m	0.451	1.282	988	999	0.487	1.379	986	999	0.513	1.455	984	998
1.0 m	0.662	1.524	980	1000	0.720	1.663	978	1000	0.758	1.761	978	998
1.5 m	0.883	1.850	969	999	0.965	2.036	964	999	1.016	2.161	961	996
2.0 m	1.204	2.354	952	996	1.339	2.663	947	996	1.422	2.867	941	989
2.5 m	1.237	2.327	948	997	1.364	2.631	944	995	1.448	2.830	941	989

Table 5.17b - ST06 Radio Path Loss Errors for Huffman Predicted Differences at 900 to 1800MHz.

Max. Err.	Frequency = 200 MHz				Frequency = 400 MHz				Frequency = 600 MHz			
	A.A.v. Error	RMSE	No. of Profs ≤ 6dB 12dB		A.A.v. Error	RMSE	No. of Profs ≤ 6dB 12dB		A.A.v. Error	RMSE	No. of Profs ≤ 6dB 12dB	
0.5 m	0.521	1.820	983	994	0.435	1.366	985	997	0.412	1.381	987	996
1.0 m	0.739	1.821	981	992	0.610	1.480	984	997	0.618	1.669	984	996
1.5 m	0.954	2.099	969	993	0.782	1.812	981	996	0.689	1.583	987	998
2.0 m	1.164	2.512	967	990	0.934	1.903	975	995	0.859	1.877	980	996
2.5 m	1.347	2.635	957	989	1.143	2.400	964	996	1.051	2.290	961	993

Table 5.18a - ST08 Radio Path Loss Errors for Huffman Predicted Differences at 200 to 600MHz.

Max. Err.	Frequency = 900 MHz				Frequency = 1400 MHz				Frequency = 1800 MHz			
	A.A.v. Error	RMSE	No. of Profs ≤ 6dB 12dB		A.A.v. Error	RMSE	No. of Profs ≤ 6dB 12dB		A.A.v. Error	RMSE	No. of Profs ≤ 6dB 12dB	
0.5 m	0.454	1.718	981	996	0.496	1.963	978	995	0.529	2.116	978	995
1.0 m	0.615	1.820	978	996	0.668	2.061	976	994	0.710	2.212	975	994
1.5 m	0.715	1.727	981	997	0.766	1.920	974	996	0.814	2.056	971	995
2.0 m	0.878	2.034	974	994	0.942	2.245	972	994	0.999	2.394	971	993
2.5 m	1.058	2.446	954	991	1.155	2.751	953	989	1.225	2.946	953	987

Table 5.18b - ST08 Radio Path Loss Errors for Huffman Predicted Differences at 900 to 1800MHz.

However, the method has shown that further data compression is achievable at a cost of introducing some small radio path loss errors. For both ST06 and ST08, data storage can be reduced by 90% or more, when compared to the original two-byte grid DEMs, whilst 95% of all profiles can be retrieved to within a radio path loss error of 6 dB. The grid reconstruction times for this method are also more efficient than the error-free models (see Section 5.3), since there are fewer codewords to be searched in the look-up table, and a greater frequency of short codes. In the above models, the grid reconstruction times for the 2.5 metre maximum error model are 80% and 57% of the times for the error free models of ST06 and ST08, respectively.

5.5 Sub-Sampled Huffman Grids

Huffman coding is at its most efficient for skewed probability distributions, such as a set of elevation differences. It has been shown that for a regular 50 metre grid DEM, the variability within terrain can be modelled efficiently by the allocation of variable length codes. However, further investigation was necessary to see whether comparable compression ratios could be achieved for data sampled at different resolutions. Huffman coding can therefore be applied to the methods discussed in Chapter Four, but may not necessarily produce the same compression ratios.

The regularly sub-sampled grid DEMs at resolutions of 100 and 200 metres were used to test the data storage compression for Huffman coding. The results of this, in terms of average codeword length in bits, and percentage of original two-byte grid storage sampled at 50 metres is shown below in Table 5.19, for both ST06 and ST08. The Huffman coding was applied to the altitude differences between the predicted and actual elevations.

Grid Cell Size	% of Points	ST06		ST08	
		Bits Per Height	% of 2-Byte Storage	Bits Per Height	% of 2-Byte Storage
50 m	100.00	1.968	12.30	3.238	20.24
100 m	25.13	2.783	4.37	4.455	7.00
200 m	6.34	3.749	1.49	5.779	2.29

Table 5.19 - Data Storage of Huffman Encoded Sub-Sampled Grids for ST06/08.

As the grid sampling density becomes sparser, the relationship between grid nodes becomes more unpredictable, resulting in a flatter and wider distribution of altitude differences. This means that the average code length for these grid DEMs will increase significantly with sampling width. In Table 5.19, as the grid cell size doubles, the average bit length of the code increases by 30 to 40% each time. By the same reasoning, it would be expected that the average

code length of a denser grid, sampled at 25 metre intervals, would produce a corresponding reduction. It can be concluded therefore, that the efficiency of Huffman-encoding grid DEMs is dependent upon both the sampling density and variability of a surface. This is to be expected, since the method has been shown to be adaptive to terrain variability. Sub-sampled grids can be considered to be a crude generalisation of dense grids and will thus exhibit a higher degree of terrain variability.

However, for the above models, the compression that is achieved by Huffman coding is accomplished without introducing any further elevation errors into the sub-sampled models. Hence the elevation and radio path loss errors in Section 4.2 of Chapter Four are the same as for these models. In that chapter, a grid sampling density of less than 100 metres was not recommended, so by just considering the 100 metre model, 96% and 93% of profiles for ST06 and ST08 can be retrieved to within a 6 dB radio path loss error with respective storage savings of 95.5% and 93%. The corresponding grid reconstruction time for these models was 1.15 and 1.85 seconds of CPU time, although profiles can be retrieved twice as fast as the original 50 m grid.

5.6 Variable Density Huffman Grids

Huffman coding can also be applied to the variable density grids of Chapter Four, although the application is not quite so straightforward. Since sub-grids may consist of 2x2, 3x3, 5x5 or 9x9 nodes, the organisation of the codes must be planned, such that the data structure does not get too complicated. Since the sub-grids are applied to 50x50 patches of 400x400 metres, the first value of each local grid at [1,1] is stored in a regular grid, which is then Huffman encoded for maximum storage efficiency. Each sub-grid is then individually Huffman encoded, from their relative altitude differences. This is accomplished in a sequence through the sub-grid, starting from the first value which can be retrieved from the sparse 400 metre Huffman grid. Alternatively, the values could be predicted using the extrapolation method discussed earlier, but the benefits would not be much greater than for the relative differences, especially if there are a significant number of small sub-grids. Table 5.20 below, shows the relative storage costs as a percentage of dense regular grid storage for these Huffman encoded variable density grids.

Maximum Error Tol.	ST06		ST08	
	% Points of 50m Grid	% of 2-Byte Storage	% Points of 50m Grid	% of 2-Byte Storage
2.5 m	50.82	12.22	107.59	32.52
5.0 m	27.15	7.47	72.16	23.50
7.5 m	17.45	5.17	46.11	16.16
10.0 m	12.44	3.90	30.57	11.40

Table 5.20 - Percentage of Grid Storage for Huffman Encoded Variable Density Grids of ST06/08.

The storage compression of variable density grids is not as great as for regular grids, since the varied sampling intervals and its disjointed nature is not as amenable to Huffman coding. However, the storage savings by Huffman encoding the variable density grid DEMs are significantly better than using fixed, two byte elevation storage. The elevation and radio path loss errors associated with these models are shown in Tables 4.9 to 4.12, for both ST06 and ST08. The radio path loss results, averaged over all frequencies, show that the 7.5 metre error constrained variable density grid will produce over 95% of profiles to within a corresponding radio path loss of 6 dB or less, with a storage reduction of nearly 95% and 84%, for ST06 and ST08, respectively. These grids can be retrieved in under two seconds of CPU time, but profile interpolation may be performed in a correspondingly faster time than for the dense regular grid, if a suitable algorithm is used (eg. interpolation of profile points only at the intersection with grid cell boundaries).

5.7 Chapter Summary

Data compression of grid DEMs has shown that considerable storage savings can be attained, and in particular, the application of Huffman coding is possibly the optimal or most efficient way of achieving this. Grid elevations are not directly stored, since it is the relative differences that are the values to be compressed. Huffman coding is applied to the distribution of these differences, such that the narrower the distribution then the greater the compression ratio. Since the efficiency of Huffman coding cannot be improved upon significantly, further compression can only be achieved by an improvement in the function used to predict neighbouring grid nodes, for which the errors or differences are stored. Of the prediction functions tested in this study, the simple linear extrapolation method (discussed in Section 5.3, p.89) gives the most satisfactory results. However, whilst more complex or detailed functions may produce a smaller distribution of error corrections, it is not envisaged that such functions will significantly improve upon the compression rates achieved in this study. A summary of the grid compression techniques is presented below in Table 5.21.

In Table 5.21, all the results have been quoted in Sections 5.2 to 5.6, apart from the radio path profiles, which are taken as an average for all examined frequencies in the range of 200 MHz to 1800 MHz. The storage and RMSE figures are self-explanatory, whilst the Error Constrained column refers to methods in which the grid node elevations are constrained to within a predetermined maximum absolute error.

Grid Compression Method		% of 16-bit Storage		R.M.S.E. (m)		Height Error	% Profiles < 6dB	
		ST06	ST08	ST06	ST08	Constrained	ST06	ST08
Differential Altitude Grids	2-bit	16.00	16.00	1.43	3.34	×	96.7	94.2
	3-bit	24.00	24.00	0.77	2.28	×	97.6	95.5
Predicted Huffman Encoding (Tolerance)	0.0 m	12.30	20.23	0.00	0.00	✓	100.0	100.0
	0.5 m	9.81	15.31	0.40	0.50	✓	98.8	98.2
	1.0 m	8.91	12.73	0.66	0.82	✓	98.2	98.0
	1.5 m	8.48	11.49	0.90	1.12	✓	97.0	97.7
	2.0 m	8.19	10.78	1.17	1.44	✓	95.6	97.3
	2.5 m	7.98	10.34	1.37	1.71	✓	95.5	95.7
Sub-Sampled	100 m	4.37	7.00	0.99	1.82	×	95.9	93.3
Huffman	2.5 m	12.22	32.52	0.42	0.33	✓	99.2	99.7
Variable	5.0 m	7.47	23.50	0.99	1.03	✓	97.8	98.2
Density	7.5 m	5.17	16.16	1.59	1.88	✓	95.5	95.6
(Max.Error)	10.0 m	3.90	11.40	2.13	2.73	✓	93.0	93.4

Table 5.21 - Summary of Grid Compression Methods for ST06 and ST08.

The results presented in this chapter, together with examination of this table has led to the following conclusions:

- (i) Models that are not error constrained cannot ensure a satisfactory surface fit, since there is no control over range of elevation errors, with a corresponding unpredictable performance in radio path loss estimation. Even though such methods may have what seems an 'acceptable' elevation RMSE, compared to constrained models, path loss error may be significantly worse (cf. 100 m sub-sampled grid with variable density grid at 5 m for ST06 and 7.5 m for ST08).
- (ii) In general, sub-sampled or variable density grids do not give the same performance in radio path loss error as for regular grids of comparable storage. This can be shown from the above table by comparing the variable density grid with the regular predicted grids. For this reason, the compressed regular grid can be considered better than the compressed variable density grid for ST08. Whilst the comparison for ST06 is satisfactory, there are significant differences for ST08, since the storage difference between the two data sets is greater for the variable density grid.
- (iii) The differential altitude grid produces satisfactory, fairly uniform path loss results, irrespective of whether two or three-bit difference codes are used to represent the step in elevation between neighbouring grid nodes. This is unusual, since there is a noticeable difference in elevation RMSE between the two models. This again suggests that the use of a dense regular grid of elevations, such as the differential altitude grid, is very suitable for radio path loss calculations, despite some large elevation errors.

- (iv) The Huffman-encoded regular grid can be considered to give the best overall results in terms of elevation errors and radio path loss performance. Elevation errors can be constrained to maximum limits (or no error at all), whilst storage savings approaching 90% can be achieved with over 95% of profiles within 6 dB path loss error.

The above conclusions do not consider profile retrieval time from the compressed models. Results have shown that retrieval from compressed, partitioned structures is rather unsatisfactory in comparison with that from an ordinary 2-byte regular grid. However, grid decompression times for the total surface (before profiling) are favourable if a number of radio path profiles are required from a single model. Results have shown that this can be accomplished for a 'one-off' overhead of 3 to 5 seconds to construct the 50 metre regular grids of ST06 and ST08, with each individual profile interpolated in an average time of less than 0.3 milliseconds.

Chapter Six

Polynomial Surface Patch Modelling

6.1 Introduction

Statistical and mathematical techniques of surface representation are widespread within the geosciences, and in particular the application of trend surface analysis. Trend surface analysis

"comprises a series of techniques for filtering data; in most instances a measured variable (z) is assumed to be a dependent variable with respect to the geographic, spatial, independent variables x and y. The initial objective of the analysis is to use a series of filtering processes to identify the function $z=f(x,y)$ that isolates and represents (1) the essential features of the regional variation pattern of z and (2) the local and the error components ('noise') included in actual observations of z. It is an objective, quantitative, descriptive technique" (Whitten, 1975).

A review of some of these methods is given in Chapter Two. However, the representation of surfaces by a compact mathematical expression, preferably a polynomial (or power series), is clearly the best (Pfaltz, 1975), since any continuous surface can be approximated with arbitrarily small error by a polynomial of sufficiently high degree. In addition, mathematical modelling of local surface geometry using locally valid surfaces has the advantage that only local data need be processed and the complexity of the mathematical model can be held to a reasonable level (Junkins et al, 1973). Hence, a polynomial DEM based on local surface patches was developed in order to evaluate the computational and storage efficiency of such a system, together with the determination of the optimal polynomial function and range of validity.

6.2 Polynomial Regression

Polynomial regression attempts to model a surface using a least squares 'best-fit' technique through the observed data points. The errors associated with each vertex are independent of each other with mean zero and variance σ^2 , and represent unpredictable fluctuations in surface values from point to point (Cliff et al, 1975). The general linear model incorporated is basically the same for any degree of surface and number of data points involved. For example, consider the first-order surface (linear plane):

$$z_{\text{trend}} = a + bx + cy, \quad \dots [6.1]$$

where a represents the elevation of the plane surface at the origin ($x=0, y=0$) and b and c are the gradients in the x and y directions respectively. These coefficients are computed such that the sum of the squared deviations is a minimum. Hence,

$$F(a,b,c) = \sum (z_{\text{observed}} - a - bx - cy)^2, \quad \dots [6.2]$$

which if it is to be minimised, must have the conditions:

$$\delta F / \delta a = \delta F / \delta b = \delta F / \delta c = 0. \quad \dots [6.3]$$

These partial derivatives are:

$$\begin{aligned} \delta F / \delta a &= \sum 2(z_{\text{observed}} - a - bx - cy)(-1) = 0 \\ \delta F / \delta b &= \sum 2(z_{\text{observed}} - a - bx - cy)(-x) = 0 \\ \delta F / \delta c &= \sum 2(z_{\text{observed}} - a - bx - cy)(-y) = 0 \end{aligned} \quad \dots [6.4]$$

which can be reduced to a set of three simultaneous equations, or normal equations:

$$\begin{aligned} an + b \sum x + c \sum y &= \sum z \\ a \sum x + b \sum x^2 + c \sum xy &= \sum zx \\ a \sum y + b \sum xy + c \sum y^2 &= \sum zy \end{aligned} \quad \dots [6.5]$$

where n is the number of data points and $z = z_{\text{observed}}$. These equations may be rewritten as a single matrix equation:

$$\begin{bmatrix} n & \sum x & \sum y \\ \sum x & \sum x^2 & \sum xy \\ \sum y & \sum xy & \sum y^2 \end{bmatrix} \times \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \sum z \\ \sum zx \\ \sum zy \end{bmatrix} \quad \dots [6.6]$$

such that the coefficients can be solved as:

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} n & \sum x & \sum y \\ \sum x & \sum x^2 & \sum xy \\ \sum y & \sum xy & \sum y^2 \end{bmatrix}^{-1} \times \begin{bmatrix} \sum z \\ \sum zx \\ \sum zy \end{bmatrix} \quad \dots [6.7]$$

Higher degree surfaces are fitted in a similar manner, with additional normal equations for each polynomial term. For example, the second order normal equations can be rewritten as:

$$\begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix} = \begin{bmatrix} n & \sum x & \sum y & \sum x^2 & \sum xy & \sum y^2 \\ \sum x & \sum x^2 & \sum xy & \sum x^3 & \sum x^2y & \sum xy^2 \\ \sum y & \sum xy & \sum y^2 & \sum x^2y & \sum xy^2 & \sum y^3 \\ \sum x^2 & \sum x^3 & \sum x^2y & \sum x^4 & \sum x^3y & \sum x^2y^2 \\ \sum xy & \sum x^2y & \sum xy^2 & \sum x^3y & \sum x^2y^2 & \sum xy^3 \\ \sum y^2 & \sum xy^2 & \sum y^3 & \sum x^2y^2 & \sum xy^3 & \sum y^4 \end{bmatrix}^{-1} \times \begin{bmatrix} \sum z \\ \sum zx \\ \sum zy \\ \sum zx^2 \\ \sum zxy \\ \sum zy^2 \end{bmatrix} \quad \dots [6.8]$$

for the polynomial function:

$$z_{\text{trend}} = a + bx + cy + dx^2 + exy + fy^2. \quad \dots [6.9]$$

The form of the general polynomial equation is:

$$z_{\text{trend}} = a_{00} + a_{10}x + a_{01}y + a_{20}x^2 + a_{11}xy + a_{02}y^2 + a_{30}x^3 + a_{21}x^2y + a_{12}xy^2 + a_{03}y^3 + a_{31}x^3y + a_{22}x^2y^2 + a_{13}xy^3 + a_{32}x^3y^2 + a_{23}x^2y^3 + a_{33}x^3y^3 + \dots + a_{mn}x^m y^n. \quad \dots [6.10]$$

In trend surface analysis it is essential to express the 'goodness or percentage of fit' of the surface function and to determine whether this is statistically significant. The most commonly used measure is provided by the percentage reduction in total corrected sum of squares accounted for by the fitted surface, which is given by:

$$\text{Goodness of Fit} = \frac{\sum Z_{\text{trend}}^2 - (\sum Z_{\text{trend}})^2 / n}{\sum Z_{\text{observed}}^2 - (\sum Z_{\text{observed}})^2 / n} \times 100\% \quad \dots [6.11]$$

(The significance of this fit is commonly assessed by comparing the sums of squares due to the trend with those due to the residuals as an F-ratio statistical test. This estimates whether the amount of variance taken up by the regression differs significantly from that expected for an equivalent number of points with the same degrees of freedom, drawn from a random sample).

The adaptability of least squares regression is illustrated overleaf in Figures 6.1 and 6.2, for polynomials of degree one (linear plane) to ten, applied to a 50 x 50 (ie. 2.5 x 2.5 kilometres²) regular grid subset of ST08. The elevation range of this subset is 32 to 263 metres, with 51 local maxima (peaks) and 141 local minima (pits). The goodness of fit of these surfaces is tabulated below in Table 6.1, together with the number of coefficients and surface extrema (ie. local maxima and minima) for each polynomial. This table also includes the statistics for the polynomial of degree zero (ie. a level plane of average elevation).

Polynomial Degree	No. of Coeffs.	Surface Extrema	% Goodness of Fit	Largest Error (m)
0	1	0	66.357	- 175.25
1	3	0	68.200	- 127.26
2	6	1	82.623	+101.12
3	10	4	83.784	+ 97.78
4	15	9	90.505	- 58.45
5	21	16	90.889	+ 47.89
6	28	25	92.792	- 39.00
7	36	36	93.611	- 33.34
8	45	49	94.218	- 31.56
9	55	64	94.884	- 25.87
10	66	81	95.551	- 22.52
Original		192	100.00	

Table 6.1 - Polynomial Surface Characteristics for Degrees 0 to 10 and % Goodness of Fit for the 50x50 grid DEM illustrated in Figure 6.1.

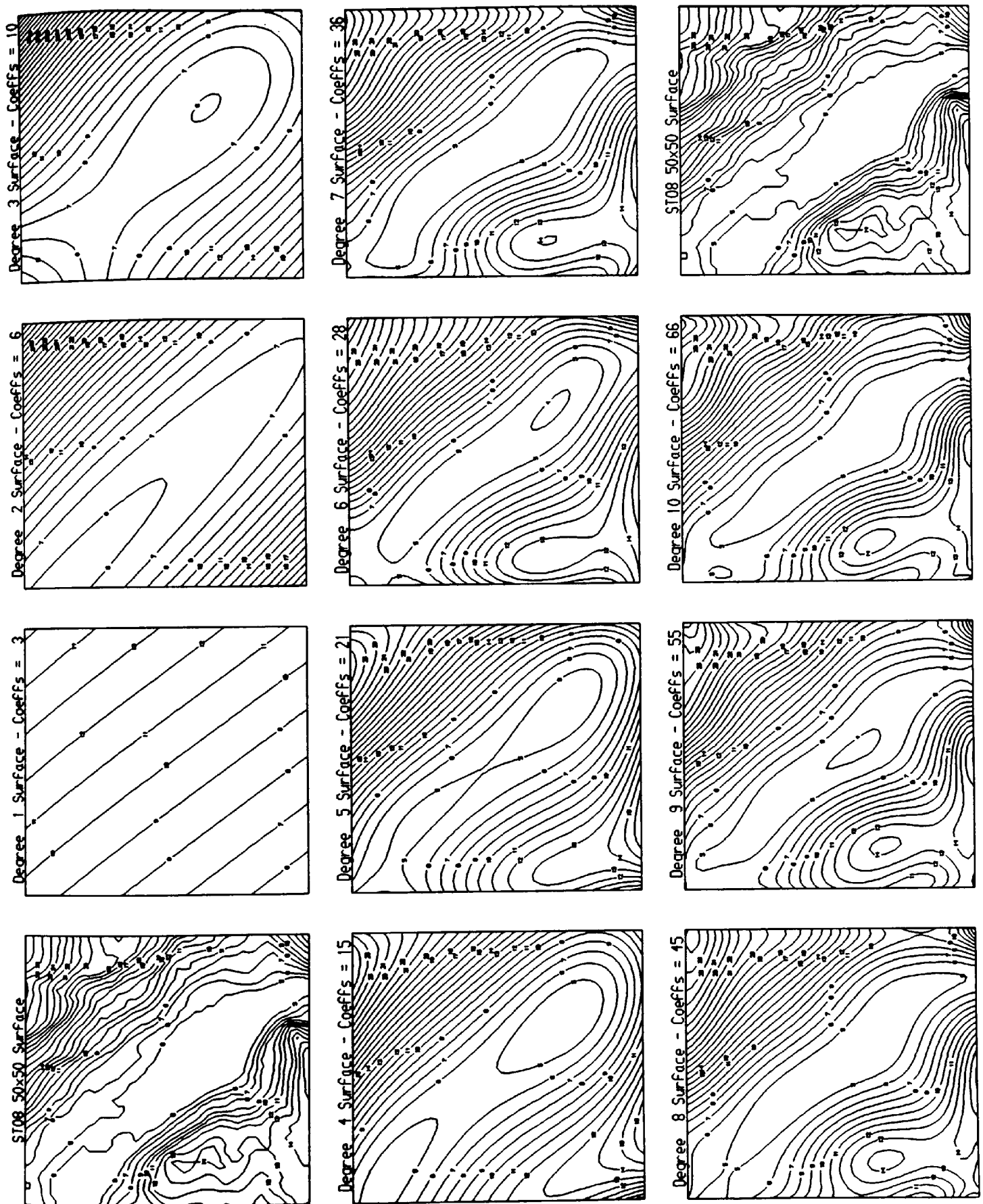


Figure 6.1 - Contour Maps of Polynomial Surfaces of Degree 1-10 and Original 50x50 Grid DEM. (Contours at 10 Metre Intervals, where Elevation = (Label - 1) x 10 metres).

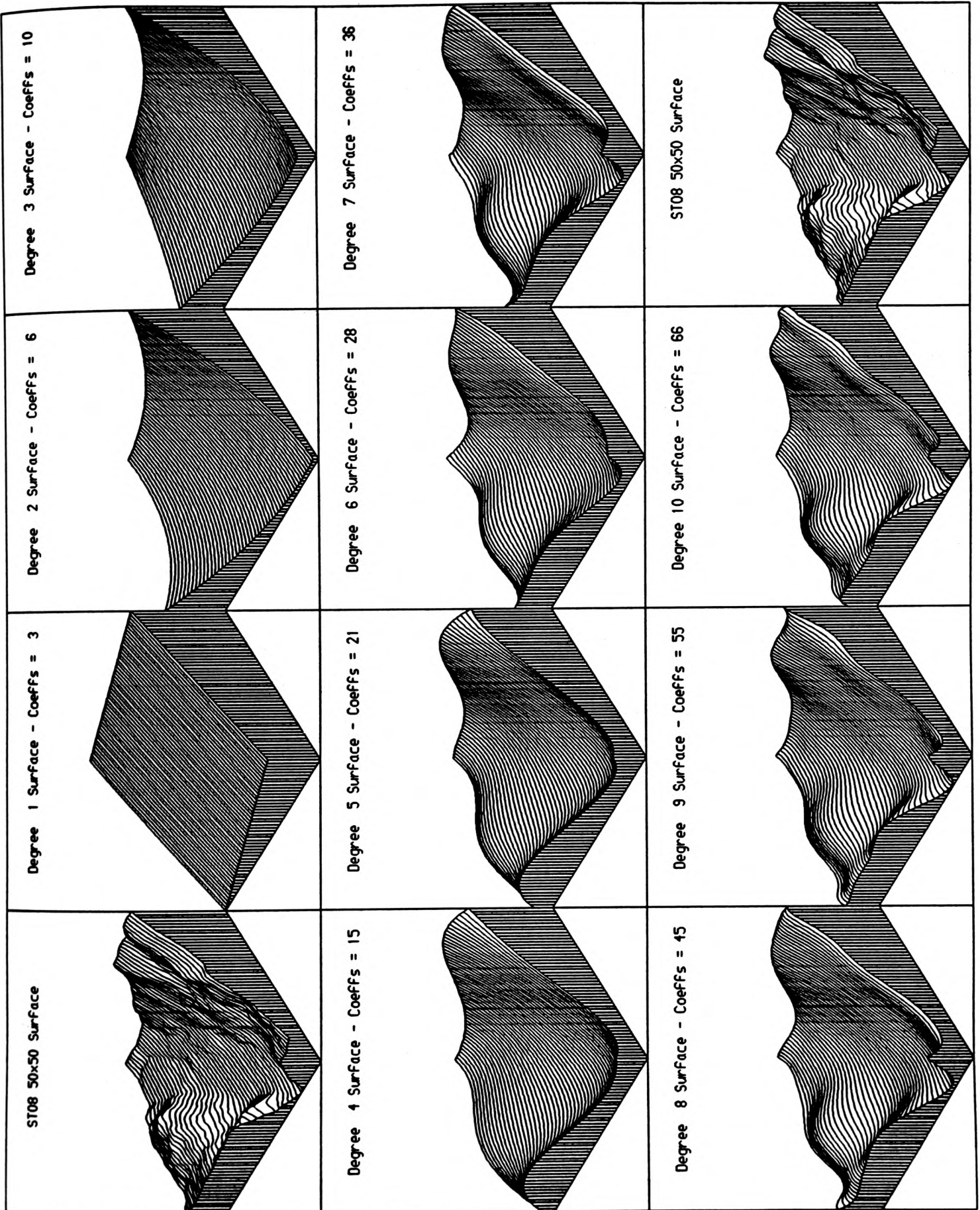


Figure 6.2 - Isometric Projections of Polynomial Surfaces of Degree 1-10 and Original 50x50 Grid DEM
(View is From the South East Corner of Contour Maps in Figure 6.1).

Figures 6.1 and 6.2 illustrate the refinement of the polynomial model to produce a better surface fit at each stage. The goodness of fit estimates in Table 6.1 show that 'good' performances are achieved by the functions, even though the 2500 data points are modelled with a maximum of only 66 coefficients (at degree 10). However, Figures 6.1 and 6.2 highlight some of the fundamental weaknesses of polynomial modelling. It is apparent from the isometric projections that the polynomial models are 'too smooth' in relation to the original surface. In reality, terrain never behaves in this manner, since it is variable with many local fluctuations. Polynomials only identify the major trends in the data, and as such cannot model to any great extent the local maxima and minima or breaks. The maximum number of surface extrema that can be modelled by a polynomial of degree n is given by $(n-1)^2$. The contour maps also illustrate the smooth nature of the polynomial surfaces, together with an inability to accurately model steep slopes. For example, valley sides are shown as contours with a fairly uniform spacing. It is in these regions that the largest elevation errors are likely to occur.

Despite these limitations, polynomials offer substantial storage savings over the regular grid DEM. The coefficients of the approximating polynomials may be stored as 16 or 8 byte floating point numbers for greater accuracy, but for the purposes of this work, four-byte values were considered. As such, the relative storage costs of each polynomial coefficient is equivalent to two grid elevations. Hence the polynomial functions illustrated in Figures 6.1 and 6.2 produce storage savings in excess of 94.7% (10th order). This highlights the storage-efficient capability of polynomials in identifying the major terrain characteristics. The problems associated with the method, such as over-smoothing or generalisation, misrepresentation of steep slopes and an ignorance of local variability needed further examination, especially with regard to its effect on radio path loss errors. The example used in this section is only an 'exaggerated' test model, whereas for a more feasible terrain model, the surface patch would be significantly smaller. However, it does illustrate the fact that polynomials offer a storage-efficient means of modelling terrain characteristics.

6.3 Polynomial Surface Patch DEM

The polynomial surfaces illustrated in Figures 6.1 and 6.2 show that in order to achieve a 'satisfactory' fit which closely approximates the original surface of the terrain, the degree of the polynomial may be inordinately high. This assumes that the polynomial achieves more than merely identifying the underlying surface trend. As such, a percentage goodness of fit is not necessarily a reliable statistic or criterion for surface modelling, with respect to an estimate such as maximum absolute error. For example, the elevation errors in the example above may be considered intolerable for many digital terrain modelling applications.

A simple approach which keeps the degree of the approximating polynomial down to a manageable (and acceptable) size is to limit the domain of validity of the expression (Pfaltz, 1975). A possible solution is to subdivide the domain of the surface into many small regions over which the surface cannot be too irregular and create a collection of approximating expressions that accurately represent the surface (Pfaltz, 1975). Such a terrain model is described by Junkins et al (1973) and elaborated upon by Pfaltz (1975). A major problem encountered in such a representation is ensuring that the surface is continuous (and smooth) along the common boundaries of the patches. This is essential for applications such as contouring. Junkins et al (1973) describe a simple method to achieve this, by the multiplication of preliminary, but overlapping approximations by appropriate weighting functions, which yield surfaces of cartographic quality. However, the storage (or computational) efficiency of this DEM is hindered by enforcing the level of smoothness (using weighting functions of various degree). Similarly, the overlapping patches can be considered inefficient, since it creates a form of data redundancy.

A prototype surface model was developed using digitised 'critical' points, including channels, ridges, spot heights and contours. However, the imposition of a domain of validity (based upon a sparse regular rectangular grid), with or without overlapping patches creates exorbitant errors in regions of sparse data, due to the extrapolation of values near the boundaries. In such instances of relatively few or no control points, there are almost no constraints on the form of the surface at the edges. If the data is sparsely distributed, these 'edge effects' will also occur in overlapping or blended patches, such that the degree of overlapping required to produce a satisfactory fit cannot be determined without prior detailed analysis. As such, it is essential that in any polynomial surface patch DEM, the method is applied to an orderly distribution of elevations, such as a regular grid DEM.

The effect of fitting polynomial patches to regular grid DEMs was analysed with respect to blended patches using polynomial weighting functions (Pfaltz, 1975), unconstrained overlapping patches and unconstrained adjacent patches. Whilst the latter two approaches do not ensure surface smoothness or even continuity, the tests suggested that the resulting small increase in overall elevation error was acceptable when compared to the level of storage savings attained. Furthermore, the results also indicate that the most storage efficient method fits polynomial functions to distinct adjacent (non-overlapping) surface patches, in which the elevations of the regular grid are assumed to represent the heights at the centre of each grid cell, rather than at the more conventional grid vertices (Figure 6.3). Typical results show that the former approach will, in general, save in excess of 15% of the storage costs of the latter at the expense of reducing by 2% the number of re-interpolated vertices being within an absolute error of ten metres.

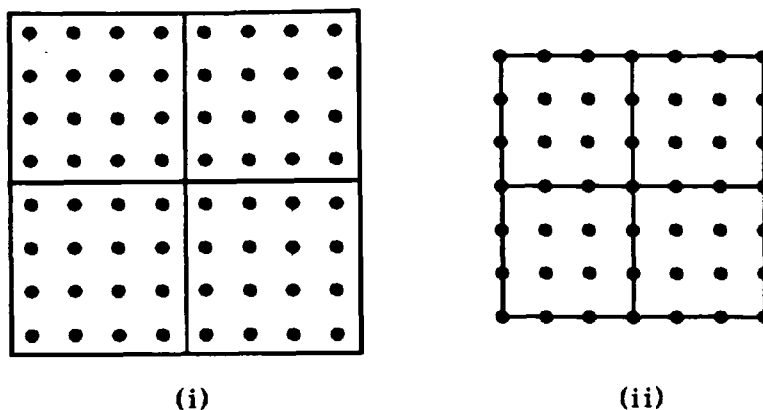


Figure 6.3 - Adjacent Polynomial Surface Patches Where Grid Values Represent Elevations at
(i) Centre and (ii) Vertices of the Cells.

The algorithm used to construct the polynomial surface patch DEM is described overleaf in Figure 6.4 and is based upon the general polynomial theory outlined in Section 6.2. This algorithm works for both regular and irregularly sampled data, but for a regular grid DEM the efficiency can be improved significantly if the data within each patch are considered locally. As such, the matrix A (which contains the sums, sums of powers and sums of cross-products of x and y within the matrix equation $AC=B$, ie. in equation [6.6]) need only be calculated once and stored for the calculation of each patch's coefficients.

The data structure used to store the polynomial coefficients is a simple matrix or grid, such that each surface patch has a unique address related to coordinate position. Hence, at the application stage, any polynomial surface can be directly accessed in a similar manner to that of the regular grid DEM. Therefore, this polynomial surface patch DEM is equivalent to the regular grid DEM, except that the grid is sparser and there are polynomial functions associated with each matrix element rather than one elevation. Search time is therefore minimal, whilst any necessary interpolation within the surface patch can be efficiently implemented with a simple substitution of the coordinates into the polynomial function.

In order to determine the optimum sampling size of the surface patches and the optimum choice of polynomial, a wide selection of both were tested. However, it was decided to limit the order of polynomials to a maximum degree of six, since the computational benefits depreciate as polynomial order increases. The larger number of coefficients require more processing time to compute and the floating point representation becomes more critical, especially since a maximum of four bytes of storage are assigned to each coefficient.

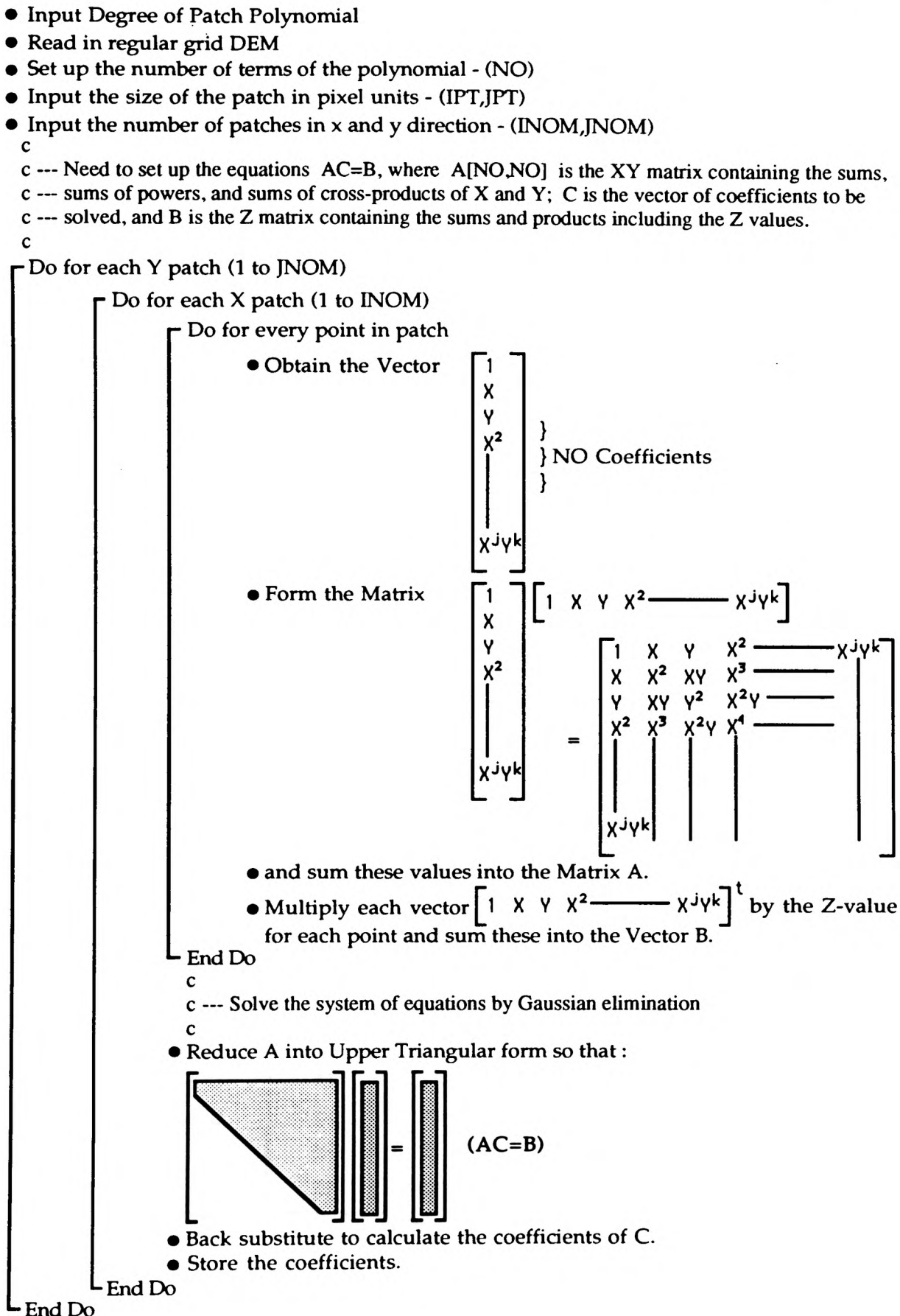


Figure 6.4 - Polynomial Surface Patch Fitting Algorithm.

The storage costs for the various models tested are outlined below in Table 6.2, where each surface model represents 400x400 regular grid elevations. The diversity of patch sizes is intended to fully test the performance and flexibility of the polynomial surfaces.

Polynomial Order	No. of Coeffs.	No. of Patches	Points per Patch	% of Grid Storage
2 a	6	80 x 80	5 x 5	48.000
2 b	6	50 x 50	8 x 8	18.750
2 c	6	40 x 40	10 x 10	12.000
2 d	6	25 x 25	16 x 16	4.688
2 e	6	20 x 20	20 x 20	3.000
3 a	10	80 x 80	5 x 5	80.000
3 b	10	50 x 50	8 x 8	31.250
3 c	10	40 x 40	10 x 10	20.000
3 d	10	25 x 25	16 x 16	7.813
3 e	10	20 x 20	20 x 20	5.000
4 a	15	50 x 50	8 x 8	46.875
4 b	15	40 x 40	10 x 10	30.000
4 c	15	25 x 25	16 x 16	11.719
4 d	15	20 x 20	20 x 20	7.500
4 e	15	16 x 16	25 x 25	4.800
4 f	15	10 x 10	40 x 40	1.875
5 a	21	50 x 50	8 x 8	65.625
5 b	21	40 x 40	10 x 10	42.000
5 c	21	25 x 25	16 x 16	16.406
5 d	21	20 x 20	20 x 20	10.500
5 e	21	16 x 16	25 x 25	6.720
5 f	21	10 x 10	40 x 40	2.625
6 a	28	50 x 50	8 x 8	87.500
6 b	28	40 x 40	10 x 10	56.000
6 c	28	25 x 25	16 x 16	21.875
6 d	28	20 x 20	20 x 20	14.000
6 e	28	16 x 16	25 x 25	8.960
6 f	28	10 x 10	40 x 40	3.500

Table 6.2 - Storage Costs of Various Polynomial Surface Patch DEMs for Degree 2-6.

Figures 6.5a and 6.5b (overleaf) illustrate the ST08 fifth order polynomial surface DEM for 20x20 patches of 20x20 points. The plots show the individually contoured surface patches and the overall surface model. The resulting storage saving over the regular grid is 89.5%.

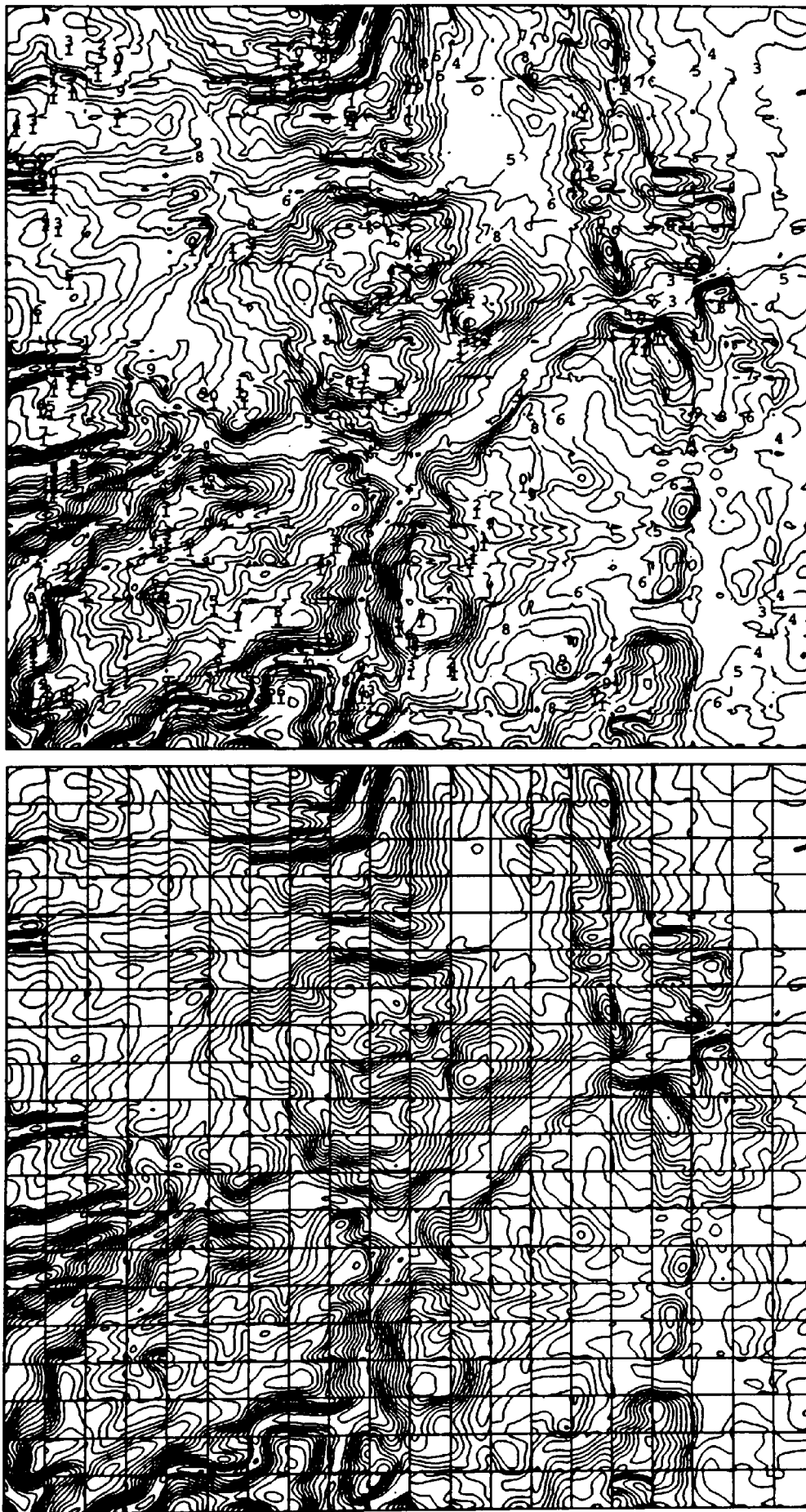


Figure 6.5a and 6.5b - ST08 Fifth Degree Polynomial DEM (20x20 Patches of 20x20 Points).

The general results in terms of elevation errors for both ST06 and ST08 are shown below in Tables 6.3 and 6.4 respectively.

Poly. Order	% of Grid Storage	Abs.Ave. Error	RMSE	Stan. Dev.	% of Interpolated Points Within Error of				
					± 5 m	± 10 m	± 15 m	± 20 m	± 25 m
2 a	48.000	0.442	0.932	1.032	99.479	99.965	99.996	100.000	—
2 b	18.750	0.830	1.715	1.905	97.506	99.671	99.936	99.979	99.993
2 c	12.000	1.112	2.249	2.509	95.423	99.198	99.808	99.949	99.989
2 d	4.688	1.933	3.605	4.091	87.983	97.073	99.186	99.799	99.940
2 e	3.000	2.415	4.352	4.978	83.767	95.180	98.563	99.584	99.873
3 a	80.000	0.292	0.602	0.669	99.884	99.998	99.999	100.000	—
3 b	31.250	0.579	1.215	1.346	98.949	99.893	99.979	99.993	99.999
3 c	20.000	0.785	1.608	1.789	97.796	99.737	99.953	99.988	99.998
3 d	7.813	1.428	2.742	3.092	92.905	98.631	99.684	99.933	99.979
3 e	5.000	1.827	3.378	3.840	89.246	97.554	99.369	99.858	99.963
4 a	46.875	0.434	0.902	1.001	99.550	99.971	99.992	99.999	100.000
4 b	30.000	0.591	1.212	1.348	98.960	99.908	99.986	99.996	99.999
4 c	11.719	1.103	2.164	2.428	95.750	99.335	99.862	99.966	99.991
4 d	7.500	1.430	2.671	3.030	93.103	98.804	99.753	99.944	99.986
4 e	4.800	1.858	3.412	3.885	88.933	97.533	99.349	99.857	99.963
4 f	1.875	3.074	5.266	6.097	78.036	92.578	97.404	99.134	99.733
5 a	65.625	0.341	0.704	0.782	99.803	99.984	99.996	99.999	100.000
5 b	42.000	0.472	0.975	1.084	99.448	99.956	99.993	99.998	100.000
5 c	16.406	0.884	1.764	1.973	97.354	99.663	99.936	99.986	99.996
5 d	10.500	1.272	2.412	2.726	94.649	99.120	99.806	99.954	99.989
5 e	6.720	1.541	2.886	3.272	92.013	98.443	99.664	99.934	99.974
5 f	2.625	3.155	5.409	6.262	77.581	92.129	97.242	99.004	99.693
6 a	87.500	0.351	0.731	0.811	99.777	99.981	99.993	99.998	100.000
6 b	56.000	0.405	0.829	0.923	99.667	99.980	99.994	99.999	100.000
6 c	21.875	1.050	2.077	2.327	96.347	99.354	99.866	99.968	99.993
6 d	14.000	1.107	2.091	2.366	95.993	99.466	99.909	99.981	99.996
6 e	8.960	1.752	3.235	3.679	90.111	97.893	99.478	99.856	99.965
6 f	3.500	2.659	4.603	5.316	81.589	94.589	98.371	99.531	99.873

Table 6.3 - ST06 Polynomial Surface Patch Elevation Errors for Degree 2-6.

Poly. Order	% of Grid Storage	Abs.Ave. Error	RMSE	Stan. Dev.	% of Interpolated Points Within Error of				
					± 5 m	± 10 m	± 15 m	± 20 m	± 25 m
2 a	48.000	1.126	1.727	2.062	97.888	99.868	99.984	99.999	99.999
2 b	18.750	2.217	3.367	4.031	88.968	98.125	99.599	99.898	99.971
2 c	12.000	3.040	4.556	5.477	81.374	95.248	98.744	99.635	99.890
2 d	4.688	5.695	8.361	10.116	60.363	82.830	92.128	96.436	98.357
2 e	3.000	7.562	10.977	13.329	49.580	74.137	86.309	92.533	95.919
3 a	80.000	0.722	1.102	1.317	99.581	99.984	99.998	100.000	100.000
3 b	31.250	1.477	2.213	2.661	95.716	99.646	99.959	99.996	100.000
3 c	20.000	2.029	3.027	3.644	90.886	98.777	99.789	99.960	99.988
3 d	7.813	3.890	5.672	6.878	73.282	91.936	97.433	99.148	99.717
3 e	5.000	5.259	7.555	9.205	62.171	84.926	93.864	97.525	99.003
4 a	46.875	1.077	1.616	1.942	98.380	99.926	99.994	99.999	100.000
4 b	30.000	1.503	2.233	2.691	95.578	99.679	99.968	99.994	99.999
4 c	11.719	2.915	4.246	5.151	82.099	96.241	99.163	99.817	99.948
4 d	7.500	3.947	5.733	6.961	72.794	91.737	97.309	99.127	99.715
4 e	4.800	5.270	7.535	9.195	62.083	84.891	93.882	97.553	99.042
4 f	1.875	9.574	13.490	16.541	40.151	65.048	79.162	87.689	92.735
5 a	65.625	0.832	1.252	1.503	99.379	99.974	99.998	100.000	100.000
5 b	42.000	1.172	1.750	2.106	97.879	99.889	99.990	99.999	100.000
5 c	16.406	2.296	3.334	4.048	88.475	98.355	99.749	99.959	99.991
5 d	10.500	3.501	5.096	6.183	76.739	94.004	98.310	99.459	99.806
5 e	6.720	4.277	6.102	7.451	69.398	90.007	96.823	99.044	99.706
5 f	2.625	9.601	13.498	16.564	39.948	64.831	79.150	87.486	92.627
6 a	87.500	0.847	1.265	1.522	99.371	99.979	99.998	100.000	100.000
6 b	56.000	0.998	1.487	1.791	98.825	99.952	99.993	99.998	100.000
6 c	21.875	2.745	4.015	4.864	84.063	96.952	99.318	99.812	99.940
6 d	14.000	2.921	4.192	5.109	82.131	96.431	99.287	99.854	99.969
6 e	8.960	4.968	7.258	8.795	64.588	86.779	94.793	97.860	99.061
6 f	3.500	7.972	11.178	13.729	45.818	71.689	85.010	92.094	95.863

Table 6.4 - ST08 Polynomial Surface Patch Elevation Errors for Degree 2-6.

The relationship between root mean square error and storage costs is illustrated overleaf in Figure 6.6 for both ST06 and ST08. The graph clearly shows a uniform relationship between error and storage, regardless of polynomial degree and patch size. This suggests that the choice of optimal polynomial and its domain of validity is discretionary. However, it is noticeable in Figure 6.6 that there are two surfaces (at approximately 9% and 22% of grid storage) in which there is a sharp increase in model error, for both ST06 and ST08. These deviations from the

norm occur for the sixth degree polynomial surfaces, suggesting that the higher degree surfaces are not as reliable as lower degree surfaces. This is partly due to the representation of coefficients as four-byte values and the oscillatory nature of higher degree surfaces. It can be concluded therefore, that a polynomial surface patch DEM should be based upon polynomials of low order (\leq degree 5) with many small patches. The computational efficiency of such a system improves as the polynomial degree decreases, since interpolation (ie. substitution into functions) is faster with fewer coefficient terms.

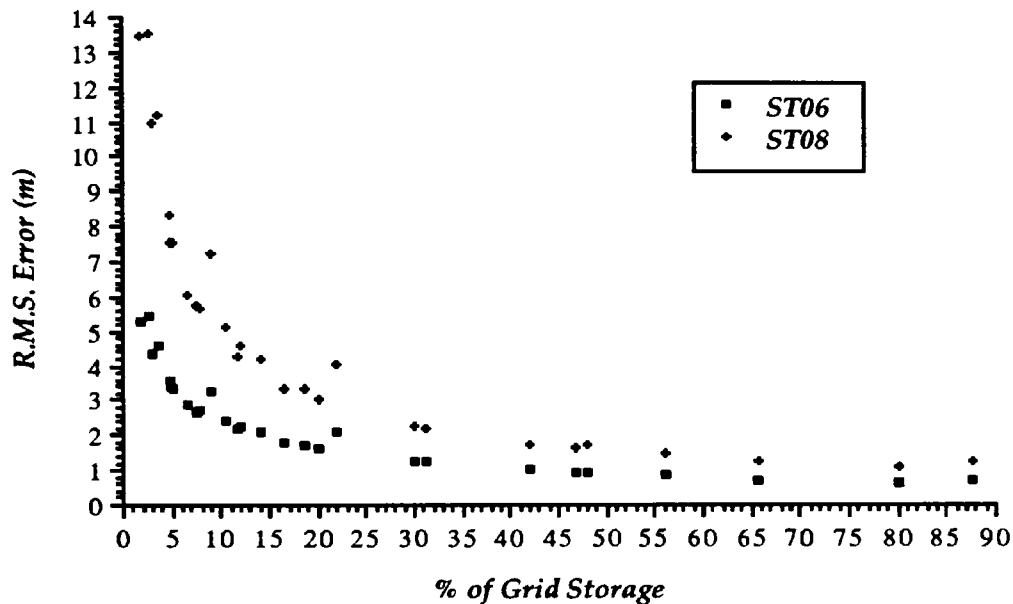


Figure 6.6 - Root Mean Square Error (m) of Polynomial Surface Patch DEMs v. % of Grid Storage.

However, this polynomial surface patch DEM is unconstrained, such that the maximum absolute error cannot be predetermined and is hence uncontrollable. Further examination of the errors within the polynomial DEMs indicate that whilst only a very small percentage of original grid points ($<0.25\%$) are greater than ± 25 metres for a typical model ($> 90\%$ storage saving), some errors can be in excess of ± 40 metres. These are likely to occur in areas of steep slope, where the polynomial is too inflexible at modelling the variation in terrain. For example, in the ST06 polynomial models, the variation in terrain along the coastline (ie. cliffs in excess of 50 metres dropping to sea level within one or two grid elements) is typical of the inflexibility of polynomial modelling. It is unreasonable to suggest that a mathematical technique of surface representation can efficiently model such breaklines. This is true of more conventional DEMs in general, since features such as cliffs are usually represented with some spatial error. For example, the vertical slope of a cliff is represented as a gradual slope in a grid DEM, the degree of which depends on the horizontal sampling interval of elevations.

Interpolation of a profile across the polynomial surface patch DEM requires efficient access to the individual functions. This is achieved by storing the polynomials in a fixed grid or matrix, such that the coordinates of any point can be mapped into a cell address. The function can then be directly retrieved and all points within that patch interpolated. Profile interpolation is carried out in a similar manner to that of grid interpolation, with points interpolated at fixed intervals (usually 50 metres). The patches intersected by each profile can be directly retrieved for interpolation. Since each patch may be used for a number of interpolations, the expense of substituting the coordinates into a number of coefficients is generally offset by having to retrieve fewer functions. As such, the process is very efficient and computationally is equivalent to that of grid DEM interpolation. Extensive tests on all the polynomial surface DEMs detailed above, resulted in profile interpolation times of between 1 and 1.2 of the regular grid interpolation time. As expected, the lowest interpolation times are for the lower degree polynomial surfaces. Hence, polynomial interpolation is equivalent to regular grid interpolation.

The largest elevation errors within the polynomial surface patch DEMs tend to occur in areas of steep slope, such as valleys. Whilst the significance of such features may be lost, the major trends within the surface will be incorporated. Therefore, the effect of elevation errors in profile interpolation for radio path loss calculations should, in theory, be minimal. Since it is very unlikely that transmitters or receivers will be sited on the slope of a steep valley, the profiles that create significant path loss errors should be constrained to those that pass through a valley (which may change from a line-of-sight to an obstructed profile), or those in which the change in slopes create significant reflection losses. The radio path loss results presented below are created using randomly generated profiles. As such, there are no constraints on the siting of the transmitter or receiver, so some profiles may have significant path loss errors. Tables 6.5a and 6.5b record the radio path loss errors for ST06 (with at least one of the transmitter/receiver endpoints being above sea level), whilst Tables 6.6a and 6.6b record the corresponding losses for ST08. These results are tabulated for each of the polynomial surface patch DEMs in Table 6.2.

Poly. Order	% of Grid Storage	Frequency = 200 MHz				Frequency = 400 MHz				Frequency = 600 MHz			
		A.Av. Error	RMSE	No. of Profs ≤ 6dB 12dB		A.Av. Error	RMSE	No. of Profs ≤ 6dB 12dB		A.Av. Error	RMSE	No. of Profs ≤ 6dB 12dB	
2 a	48.000	0.665	1.588	989	996	0.694	1.793	982	997	0.608	1.508	985	999
2 b	18.750	1.149	2.162	977	996	1.209	2.547	955	995	1.142	2.318	956	994
2 c	12.000	1.576	2.838	955	992	1.665	3.295	937	985	1.625	3.143	936	987
2 d	4.688	2.488	3.938	884	985	2.529	4.486	877	967	2.496	4.391	873	964
2 e	3.000	2.674	4.331	882	978	2.837	4.878	859	955	2.863	4.818	855	965
3 a	80.000	0.453	1.258	989	999	0.458	1.410	987	997	0.423	1.200	990	1000
3 b	31.250	0.882	1.837	980	996	0.898	2.067	972	994	0.817	1.854	977	996
3 c	20.000	1.206	2.485	971	992	1.292	2.977	953	986	1.182	2.528	955	994
3 d	7.813	1.927	3.190	932	991	1.998	3.693	916	979	2.109	3.443	912	985
3 e	5.000	2.198	3.704	916	986	2.292	4.041	889	980	2.313	4.092	884	979
4 a	46.875	0.615	1.414	992	997	0.659	1.677	986	997	0.591	1.581	984	998
4 b	30.000	0.847	1.870	987	995	0.851	2.002	977	995	0.797	1.798	977	998
4 c	11.719	1.457	2.618	962	995	1.523	3.100	947	987	1.398	2.785	946	990
4 d	7.500	1.798	3.052	940	992	1.901	3.598	922	981	1.838	3.413	919	986
4 e	4.800	2.247	3.637	903	989	2.403	4.193	882	974	2.393	4.152	882	975
4 f	1.875	3.116	4.736	852	974	3.304	5.434	815	949	3.431	5.560	807	940
5 a	65.625	0.490	1.185	991	1000	0.525	1.393	989	998	0.465	1.227	989	1000
5 b	42.000	0.738	1.637	985	998	0.769	1.819	976	997	0.734	1.698	981	998
5 c	16.406	1.263	2.364	968	995	1.309	2.803	950	993	1.178	2.468	954	993
5 d	10.500	1.739	3.245	946	985	1.794	3.535	930	982	1.717	3.280	929	989
5 e	6.720	1.935	3.155	927	996	2.028	3.647	904	982	2.008	3.630	907	982
5 f	2.625	3.192	4.843	834	964	3.394	5.518	809	942	3.506	5.579	791	949
6 a	87.500	0.507	1.322	989	999	0.557	1.537	983	997	0.492	1.305	987	1000
6 b	56.000	0.571	1.380	993	997	0.613	1.522	986	999	0.555	1.324	991	1000
6 c	21.875	1.593	2.828	950	992	1.625	3.186	936	984	1.551	2.972	943	989
6 d	14.000	1.581	2.841	951	992	1.668	3.335	933	987	1.575	3.067	937	989
6 e	8.960	2.262	3.712	902	990	2.323	4.151	896	975	2.243	4.021	887	977
6 f	3.500	2.939	4.548	864	971	2.925	4.836	852	966	2.948	4.834	835	965

Table 6.5a - ST06 Polynomial Surface Patch DEM Results for Radio Path Loss Errors
at Frequencies of 200, 400 and 600 MHz.

Poly. Order	% of Grid Storage	Frequency = 900 MHz				Frequency = 1400 MHz				Frequency = 1800 MHz			
		A.A.v. Error	RMSE	No. of Profs ≤ 6dB 12dB		A.A.v. Error	RMSE	No. of Profs ≤ 6dB 12dB		A.A.v. Error	RMSE	No. of Profs ≤ 6dB 12dB	
2 a	48.000	0.617	1.561	979	998	0.679	1.716	977	998	0.716	1.812	976	998
2 b	18.750	1.240	2.581	946	996	1.384	2.895	939	993	1.464	3.078	934	992
2 c	12.000	1.773	3.353	913	987	1.993	3.786	900	976	2.119	4.041	895	972
2 d	4.688	2.800	4.886	837	951	3.182	5.585	815	931	3.400	5.993	801	920
2 e	3.000	3.181	5.345	815	947	3.597	6.106	786	929	3.837	6.544	773	906
3 a	80.000	0.430	1.266	986	998	0.485	1.420	983	998	0.512	1.505	983	997
3 b	31.250	0.875	2.061	970	997	0.953	2.153	969	996	1.003	2.277	964	995
3 c	20.000	1.238	2.677	941	994	1.379	2.993	941	994	1.460	3.178	934	989
3 d	7.813	2.109	3.808	885	978	2.381	4.323	868	966	2.533	4.621	862	956
3 e	5.000	2.561	4.393	858	969	2.913	5.016	834	959	3.107	5.381	817	946
4 a	46.875	0.617	1.640	980	997	0.682	1.806	979	995	0.719	1.908	977	995
4 b	30.000	0.810	1.873	969	998	0.901	2.102	965	997	0.951	2.236	964	995
4 c	11.719	1.529	3.052	926	990	1.710	3.450	920	981	1.813	3.689	917	978
4 d	7.500	1.980	3.634	906	981	2.226	4.095	891	974	2.359	4.362	882	971
4 e	4.800	2.632	4.522	855	970	2.981	5.135	838	951	3.182	5.508	827	938
4 f	1.875	3.869	6.217	758	922	4.409	7.123	724	891	4.709	7.645	708	872
5 a	65.625	0.484	1.279	988	1000	0.531	1.392	986	999	0.559	1.471	985	999
5 b	42.000	0.749	1.758	973	999	0.827	1.942	972	997	0.869	2.050	971	997
5 c	16.406	1.274	2.637	940	992	1.428	2.961	931	991	1.513	3.149	928	985
5 d	10.500	1.872	3.558	903	986	2.115	4.032	895	981	2.251	4.313	887	977
5 e	6.720	2.201	3.894	885	983	2.474	4.394	866	968	2.631	4.691	858	961
5 f	2.625	3.919	6.200	744	927	4.483	7.114	714	896	4.803	7.649	703	878
6 a	87.500	0.505	1.370	982	999	0.560	1.535	980	999	0.594	1.645	980	998
6 b	56.000	0.548	1.322	989	1000	0.600	1.449	988	999	0.632	1.523	987	999
6 c	21.875	1.708	3.252	924	991	1.919	3.642	907	982	2.051	3.909	907	973
6 d	14.000	1.700	3.229	918	987	1.902	3.618	907	983	2.022	3.865	897	978
6 e	8.960	2.447	4.344	863	972	2.763	4.904	842	955	2.951	5.239	833	944
6 f	3.500	3.277	5.343	793	953	3.715	6.114	769	924	3.964	6.566	758	904

*Table 6.5b - ST06 Polynomial Surface Patch DEM Results for Radio Path Loss Errors
at Frequencies of 900, 1400 and 1800 MHz.*

Poly. Order	% of Grid Storage	Frequency = 200 MHz				Frequency = 400 MHz				Frequency = 600 MHz			
		A.Av. Error	RMSE	No. of Profs ≤ 6dB 12dB		A.Av. Error	RMSE	No. of Profs ≤ 6dB 12dB		A.Av. Error	RMSE	No. of Profs ≤ 6dB 12dB	
2 a	48.000	1.161	2.449	963	989	1.130	2.375	961	995	1.097	2.392	962	995
2 b	18.750	2.156	3.713	915	977	2.256	3.980	888	983	2.041	3.825	898	975
2 c	12.000	3.030	5.048	871	962	2.952	4.997	855	953	2.821	4.874	865	954
2 d	4.688	4.757	7.078	727	915	4.832	7.359	724	882	4.657	7.174	723	901
2 e	3.000	5.668	8.248	658	885	5.762	8.572	658	855	5.689	8.386	646	869
3 a	80.000	0.729	1.846	982	991	0.707	1.654	979	997	0.668	1.559	976	999
3 b	31.250	1.549	3.006	948	983	1.463	2.851	944	990	1.341	2.774	939	992
3 c	20.000	2.153	3.903	912	976	1.930	3.457	896	988	1.821	3.480	909	981
3 d	7.813	3.538	5.551	819	957	3.492	5.659	814	946	3.244	5.373	832	952
3 e	5.000	4.209	6.323	765	930	4.254	6.616	746	915	4.094	6.429	767	918
4 a	46.875	1.094	2.369	966	991	1.055	2.194	971	994	0.989	2.215	967	996
4 b	30.000	1.649	3.086	945	986	1.543	2.890	933	990	1.481	2.982	935	986
4 c	11.719	2.741	4.647	883	968	2.826	4.784	850	965	2.608	4.667	866	970
4 d	7.500	3.471	5.374	831	950	3.545	5.688	805	935	3.403	5.535	813	942
4 e	4.800	4.235	6.407	773	931	4.327	6.696	745	909	4.218	6.703	764	911
4 f	1.875	6.306	8.956	613	849	6.814	9.911	587	810	6.875	9.912	578	813
5 a	65.625	0.867	1.925	973	995	0.839	1.797	981	996	0.791	1.815	974	996
5 b	42.000	1.248	2.614	962	987	1.196	2.471	956	993	1.124	2.528	955	992
5 c	16.406	2.331	4.049	903	977	2.226	3.935	884	977	2.047	3.713	897	980
5 d	10.500	3.110	4.788	850	969	3.204	5.280	817	948	2.941	4.963	837	962
5 e	6.720	3.745	5.852	811	950	3.779	6.044	781	937	3.617	5.866	796	940
5 f	2.625	6.524	9.127	596	846	6.821	9.813	565	819	6.869	9.926	572	806
6 a	87.500	0.836	1.944	979	993	0.797	1.669	979	998	0.750	1.694	982	997
6 b	56.000	1.042	2.175	971	991	0.956	1.922	975	997	0.928	2.128	963	996
6 c	21.875	2.810	4.724	866	962	2.594	4.552	868	962	2.379	4.274	875	971
6 d	14.000	2.844	4.617	877	956	2.763	4.750	846	963	2.500	4.371	856	973
6 e	8.960	4.323	6.561	759	919	4.251	6.757	750	913	4.107	6.628	759	922
6 f	3.500	5.531	8.006	672	883	5.893	8.648	636	852	5.892	8.668	624	850

Table 6.6a - ST08 Polynomial Surface Patch DEM Results for Radio Path Loss Errors
at Frequencies of 200, 400 and 600 MHz.

Poly. Order	% of Grid Storage	Frequency = 900 MHz				Frequency = 1400 MHz				Frequency = 1800 MHz			
		A.Av. Error	RMSE	No. of Profs ≤ 6dB 12dB		A.Av. Error	RMSE	No. of Profs ≤ 6dB 12dB		A.Av. Error	RMSE	No. of Profs ≤ 6dB 12dB	
2 a	48.000	1.014	2.232	959	994	1.040	2.338	960	994	1.098	2.482	959	993
2 b	18.750	2.028	3.743	892	980	2.168	4.079	890	972	2.301	4.358	888	967
2 c	12.000	2.766	4.806	869	964	2.916	5.182	865	952	3.088	5.540	853	947
2 d	4.688	4.852	7.436	709	883	5.313	8.260	694	856	5.623	8.781	684	845
2 e	3.000	5.921	8.715	630	857	6.443	9.585	612	832	6.767	10.128	599	810
3 a	80.000	0.616	1.525	974	1000	0.622	1.615	975	996	0.662	1.739	975	996
3 b	31.250	1.271	2.662	944	989	1.349	2.905	947	987	1.426	3.094	945	983
3 c	20.000	1.740	3.308	914	983	1.818	3.484	913	983	1.924	3.721	907	978
3 d	7.813	3.314	5.418	822	948	3.566	5.918	808	932	3.760	6.292	796	924
3 e	5.000	4.185	6.475	751	920	4.533	7.171	739	899	4.787	7.625	728	881
4 a	46.875	0.947	2.236	967	995	0.999	2.468	963	992	1.057	2.634	963	991
4 b	30.000	1.372	2.793	934	991	1.420	2.968	937	989	1.496	3.143	933	986
4 c	11.719	2.646	4.711	861	962	2.826	5.076	858	952	2.993	5.413	848	947
4 d	7.500	3.441	5.598	808	935	3.545	6.123	789	920	3.403	6.523	781	914
4 e	4.800	4.337	6.786	749	904	4.742	7.505	740	884	5.014	7.993	727	870
4 f	1.875	7.239	10.407	563	771	7.900	11.454	552	743	8.326	12.093	538	726
5 a	65.625	0.747	1.783	973	997	0.760	1.883	974	995	0.801	2.001	972	995
5 b	42.000	1.059	2.377	952	994	1.090	2.524	957	993	1.149	2.676	953	992
5 c	16.406	2.043	3.787	897	977	2.167	4.080	898	969	2.294	4.348	894	964
5 d	10.500	2.958	4.914	832	957	3.215	5.408	821	929	3.397	5.745	815	922
5 e	6.720	3.648	5.860	790	940	3.962	6.440	776	916	4.174	6.846	765	903
5 f	2.625	7.191	10.272	554	783	7.824	11.238	540	748	8.231	11.874	535	724
6 a	87.500	0.756	1.821	973	997	0.760	1.878	973	996	0.803	1.991	974	996
6 b	56.000	0.877	2.041	965	997	0.896	2.143	965	994	0.949	2.274	963	994
6 c	21.875	2.401	4.293	874	967	2.580	4.712	863	959	2.722	5.024	857	951
6 d	14.000	2.512	4.407	864	967	2.751	4.970	852	950	2.915	5.317	890	940
6 e	8.960	4.098	6.550	749	918	4.471	7.230	735	894	4.736	7.701	727	879
6 f	3.500	6.124	9.036	610	831	6.695	9.928	595	817	7.040	10.448	584	796

*Table 6.6b - ST08 Polynomial Surface Patch DEM Results for Radio Path Loss Errors
at Frequencies of 900, 1400 and 1800 MHz.*

The relationship between radio path loss errors and storage behaves in a similar manner as that of the polynomial elevation root mean square errors (Figure 6.6). This is illustrated below (Figure 6.7) in terms of root mean square error (in dBs). It is noticeable that this similarity is reflected in the performance of the sixth degree polynomial surface patch DEMs (at 9% and 22% of grid storage), in which fluctuations from the norm are due to the larger number of coefficients and their greater significance. The performance of the number of profiles within radio path loss errors of ± 6 and ± 12 dBs is illustrated in Figure 6.8.

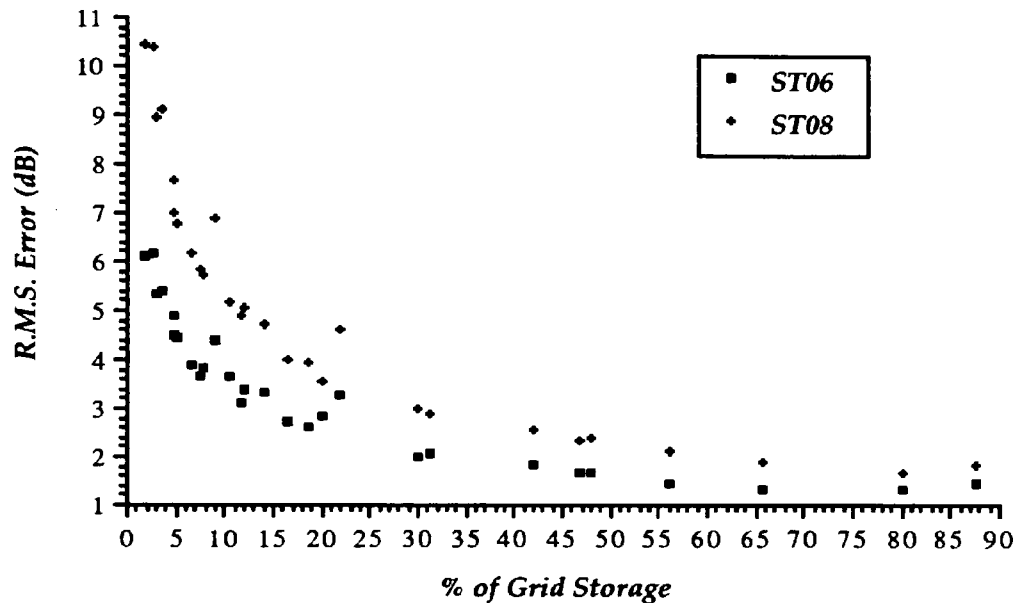


Figure 6.7 - R.M.S. Error (dBs) of Radio Path Losses for Polynomial Surface Patch DEMs Averaged Over the Frequencies of 200, 400, 600, 900, 1400 & 1800 MHz.

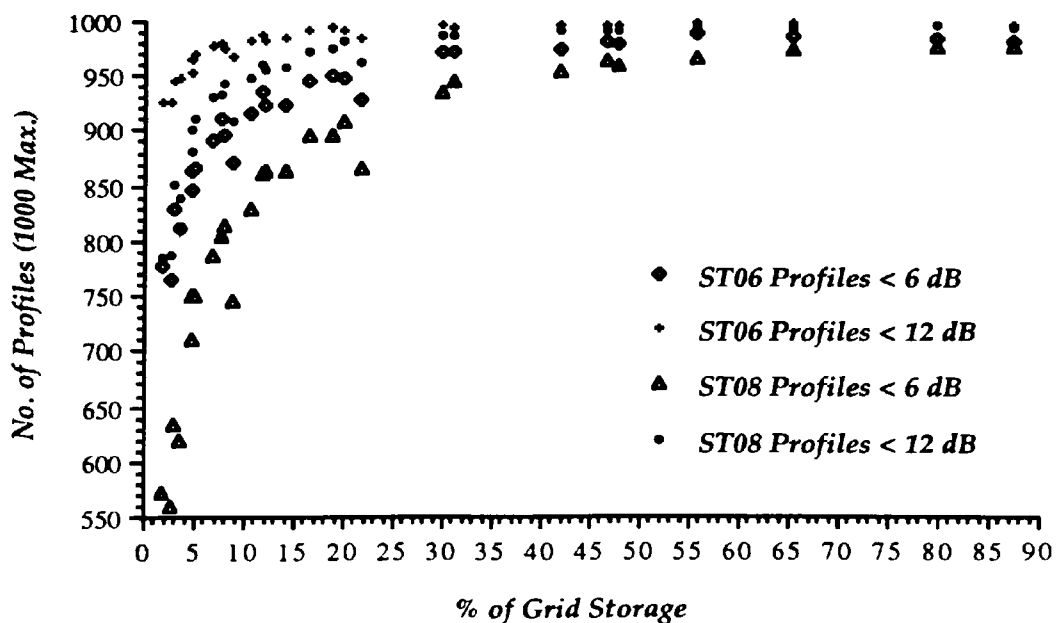


Figure 6.8 - No. of Polynomial Surface Patch Profiles Within Path Loss Error of 6 & 12 dBs Averaged Over the Frequencies of 200, 400, 600, 900, 1400 & 1800 MHz.

6.4 Polynomial Surface Patch Summary and Conclusions

Polynomial surface patch DEMs are a flexible and storage-efficient means of representing terrain. Despite their inability at adapting to some of the finer features of surfaces, such as local maxima and minima, and a misrepresentation of steep slopes, the overall performance of the polynomial models is very encouraging. By using a least-squares criterion to fit the surface, the absolute average error is at a minimum, whilst the sum of residual errors is zero. Results have shown that there is no optimal choice of patch size or polynomial degree. The errors have been shown to be related to storage costs, and as such any polynomial degree surface can be used with an appropriate patch size. However, for greater efficiency, lower degree polynomials are recommended.

Some of the limitations of this polynomial surface patch modelling should be taken into further consideration. The definition of fixed patch constraints (based on a regular grid), together with the use of polynomials of fixed degree may create 'inflexibility' within the model. In order to create a surface patch DEM of 'good' performance, the degree of the polynomial used to model the patches may be inefficient in many cases. The division of regular grid DEMs into surface patches creates individual surfaces which should be considered independent of one another. As such, the limitations of using a fixed degree polynomial for each surface patch creates data redundancy, in that polynomials of lower degree could most probably be used for some patches. In effect, this data redundancy can be considered to be of the same form within a regular grid DEM. The fixed constraints of polynomial degree and patch size do not allow an ability for the polynomials to adapt themselves to the terrain variability. For example, in the polynomial surface patch DEMs of ST06, the patches at sea level are represented with the same number of coefficients as in areas of variable terrain.

There are two possible solutions to this problem. The first assumes that the degree of the approximating polynomial within each surface patch is variable. This allows greater flexibility for surface modelling since patches can be considered independent of one another. The variability within each patch may then be modelled by the polynomial of lowest degree which constrains the fit of the surface to some predetermined tolerance. Hence, in ST06, the surface patches at sea level could be represented by polynomials of degree zero (ie. one coefficient representing an average elevation of 0 metres). This approach is considered in Section 6.5, below. Jancaitis (1978) uses a four coefficient bilinear polynomial for terrain data compaction, but also concludes that the length of the polynomial should be automatically varied to achieve the desired goodness of fit. The second approach is to vary the size of the polynomial's domain, rather than varying the polynomial degree. This can be accomplished with a quadtree data structure and is considered further in Chapter Seven. Both approaches not only model the surface data more efficiently, but allow constraints to be imposed on the

DEM construction, such that the degree of error is controllable. This makes surface modelling using polynomials more viable and attractive as a DEM.

6.5 Adaptive Polynomial Surface Patch DEM

This approach is more flexible than the fixed polynomial surface patch DEM, since the degree of polynomial surface used to represent the fixed size patch is dependent upon the nature of the terrain. The surface of each patch is modelled by polynomials of increasing degree, until the maximum absolute error is within a predetermined tolerance. However, since some surfaces may be unsuitable for efficient polynomial modelling, an upper bound on the degree of function should be set. For this DEM, the range of polynomial order is from 10th degree surfaces of 66 coefficients down to zero degree polynomials of one coefficient. The number of coefficients for each polynomial is shown in Table 6.1 (in Section 6.2). This upper bound on polynomial order is set, since some patches may require inordinately high functions. As a result, the DEM is not completely error-constrained, since there is a likelihood that some elevations may be beyond the predetermined error tolerance. This occurs in patches of extreme terrain variability, and in particular along steep gradients, such as cliffs. In such instances, the polynomial which best fits the surface is adopted as the representative function. In most cases this is likely to be the 10th degree polynomial function.

The data structure used to represent this adaptive polynomial surface patch DEM is not as simple as the grid structure used in Section 6.3, due to the variable nature of polynomial coefficients. Since the surface patches are organised in relation to the original regular grid DEM, it is advisable to base such an indexed data structure on a grid referencing scheme. As such, a grid corresponding to the patches is used to indicate the position in a secondary data structure of the first coefficient of each surface. The number of coefficients for a surface patch is then determined from the start position of its neighbouring function. Hence, the coefficients for all the surface patch polynomials are stored in an ordered one-dimensional file, referenced by the grid. The overheads of using such an indexing scheme (in addition to storing the polynomial coefficients) is dependent upon the number of surface patches within the model, ie. for the pointers based on the grid. These pointers are stored as two-byte integers, which for the surface models considered in this section, accounts for between 0.25% and 1% of the original grid storage.

For this prototype surface model, three patch sizes were considered and examined in detail, each corresponding to 400x400 points in the original grid DEMs. These are 40x40 patches of 10x10 points, 25x25 patches of 16x16 points and 20x20 patches of 20x20 points. For each of these models, adaptive polynomial patch DEMs were constructed for surfaces with maximum

absolute errors of 10, 15, 20 and 25 metres. However, as mentioned above, in some instances the maximum errors may exceed these tolerances, due to the constraint of imposing polynomials of maximum degree (ie. 10th order). The number of patches of each polynomial function are shown below in Tables 6.7 and 6.8 for ST06 and ST08 respectively, together with the storage costs for the coefficients in relation to the original grid storage.

Surface Patches	Abs. Error Tolerance	Number of Patches of Polynomial Degree												% of Grid Storage
		0	1	2	3	4	5	6	7	8	9	10		
40x40 Patches of 10x10 Points	10m	860	326	212	112	49	20	14	1	2	--	4	7.709	
	15m	1090	336	97	48	19	6	--	3	1	--	--	4.655	
	20m	1254	267	49	23	5	--	1	1	--	--	--	3.398	
	25m	1393	170	26	9	1	1	--	--	--	--	--	2.731	
25x25 Patches of 16x16 Points	10m	262	59	74	69	59	36	23	16	8	9	10	7.436	
	15m	316	93	100	50	32	14	9	5	2	1	3	4.055	
	20m	391	117	72	27	13	9	3	2	--	--	1	2.550	
	25m	439	118	46	14	3	3	2	--	--	--	--	1.716	
20x20 Patches of 20x20 Points	10m	148	18	27	41	43	34	26	19	11	5	28	5.744	
	15m	167	41	60	56	24	19	13	4	7	3	6	4.191	
	20m	198	70	66	26	16	10	6	4	2	--	2	2.560	
	25m	248	81	35	13	12	9	1	--	1	--	--	1.591	

Table 6.7 - Number of Polynomial Surface Patches of Degree 1 to 10 for ST06.
(Total patches for each model are 1600, 625 and 400 respectively).

Surface Patches	Abs. Error Tolerance	Number of Patches of Polynomial Degree												% of Grid Storage
		0	1	2	3	4	5	6	7	8	9	10		
40x40 Patches of 10x10 Points	10 m	92	337	380	359	257	116	31	15	3	3	7	19.293	
	15 m	215	514	448	290	101	21	4	4	--	2	1	12.166	
	20 m	360	599	422	180	32	5	--	1	1	--	--	8.944	
	25 m	502	640	363	81	12	2	--	--	--	--	--	7.040	
25x25 Patches of 16x16 Points	10 m	11	17	70	72	62	102	99	72	46	29	45	20.341	
	15 m	17	83	101	106	109	108	58	24	8	2	9	11.734	
	20 m	52	107	138	133	95	69	21	3	5	1	1	8.059	
	25 m	77	142	158	132	71	34	11	--	--	--	--	6.073	
20x20 Patches of 20x20 Points	10 m	4	6	16	19	39	36	40	45	50	42	103	19.684	
	15 m	8	24	40	48	45	59	62	44	40	9	21	12.144	
	20 m	22	39	61	56	68	71	41	23	10	3	6	8.204	
	25 m	34	61	56	81	86	46	21	11	3	1	--	5.991	

Table 6.8 - Number of Polynomial Surface Patches of Degree 1 to 10 for ST08.

The distributions of polynomial functions of various degree are illustrated below in Figures 6.9 and 6.10 for 40x40 patches at the 10 metre tolerance level for ST06 and ST08 respectively.

[illegible]

x denotes a polynomial
of degree 10.

Figure 6.9 - ST06 Distribution of Adaptive Polynomial Surface Patch Functions for 40x40 Patches (Maximum Absolute Error = 10 Metres, Storage = 7.709%).

45543422332344323233443121301145412x4411
44434132223243444413352111121561227223
394243344235564422234211112111713142x213
4553214444262435314362112121131311544411
43422221244434333334111221325201543222
3333334333224253733251112122152212132323
752234237522333444x84511111332142134312
35224345343453644702111101323122212222
224324535154237552651011212233213121111
24533122414222443454123344422521314321
3534321435553354343111112114322121343111
33545314451645523421111124212121352221
33454331324332323222121234121112243421
4232443127533223122334412135221122235323
433333221244443034513412223442232343634
333433362321123224332241225343322333525
25533552556316434425456229144342232434
344423277125465524424564216343333344453
373633342354743133233242553243534225233
3353433344455243354325424435334332333
5533456346345434x431213357633283443311121
3340342445465444352233432333523265132
452232322525x32212242435374121000001132
3322224433414433123354211521001000013
33321222362111134312335211211000224211
2443124113621432134412122111210022423113
222440122151232231322021422010103244332
23133011115123221224122232332433565
1333543112312321101321223443544325644
623412122112321112123623353332554234
5624443111030242112122534544443134454222
3322534222221111112353225436444223111
422233353231234223367123244321111111
433213415434433334422142544421111111
242322120224333422223383423431112111
1213220021212026411423144443311011000
012121001100111214224224310322100011012
12231100232203102334431030302100000002
22112113132411100264332004532000000011
000120131211102012111101100000000111

x denotes a polynomial
of degree 10.

Figure 6.10 - ST08 Distribution of Adaptive Polynomial Surface Patch Functions for 40x40 Patches (Maximum Absolute Error = 10 Metres, Storage = 19.293%).

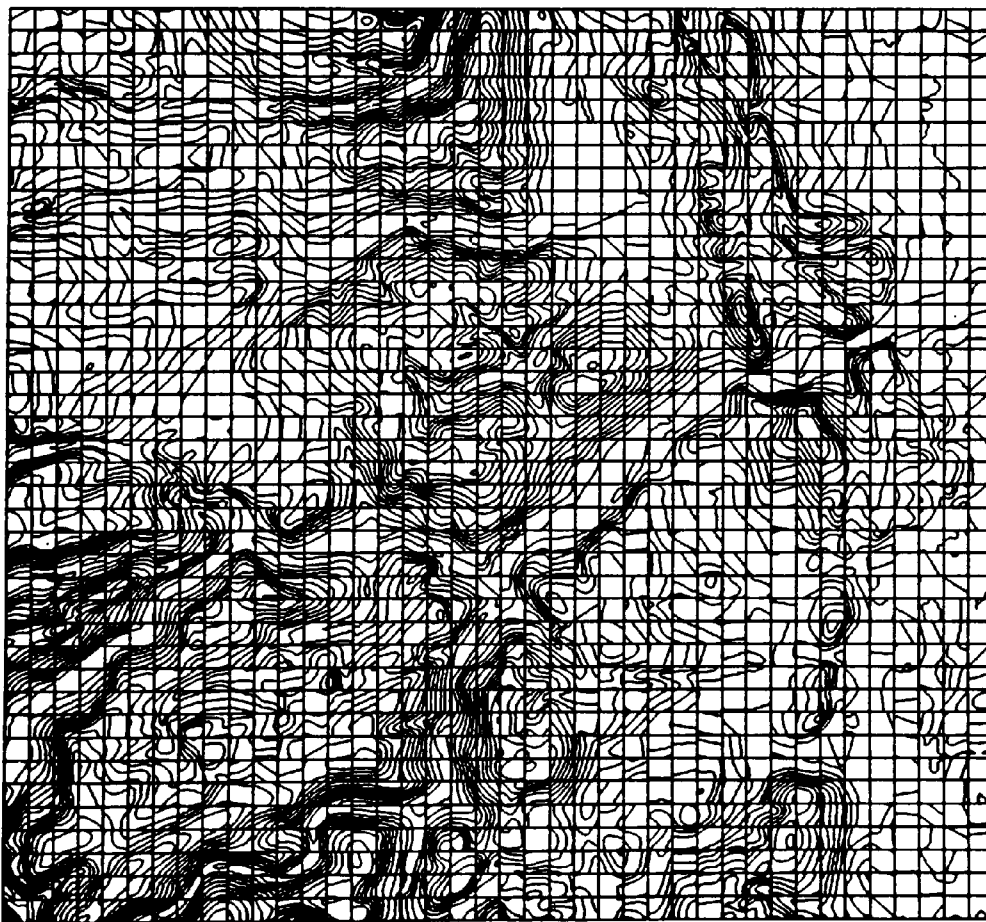
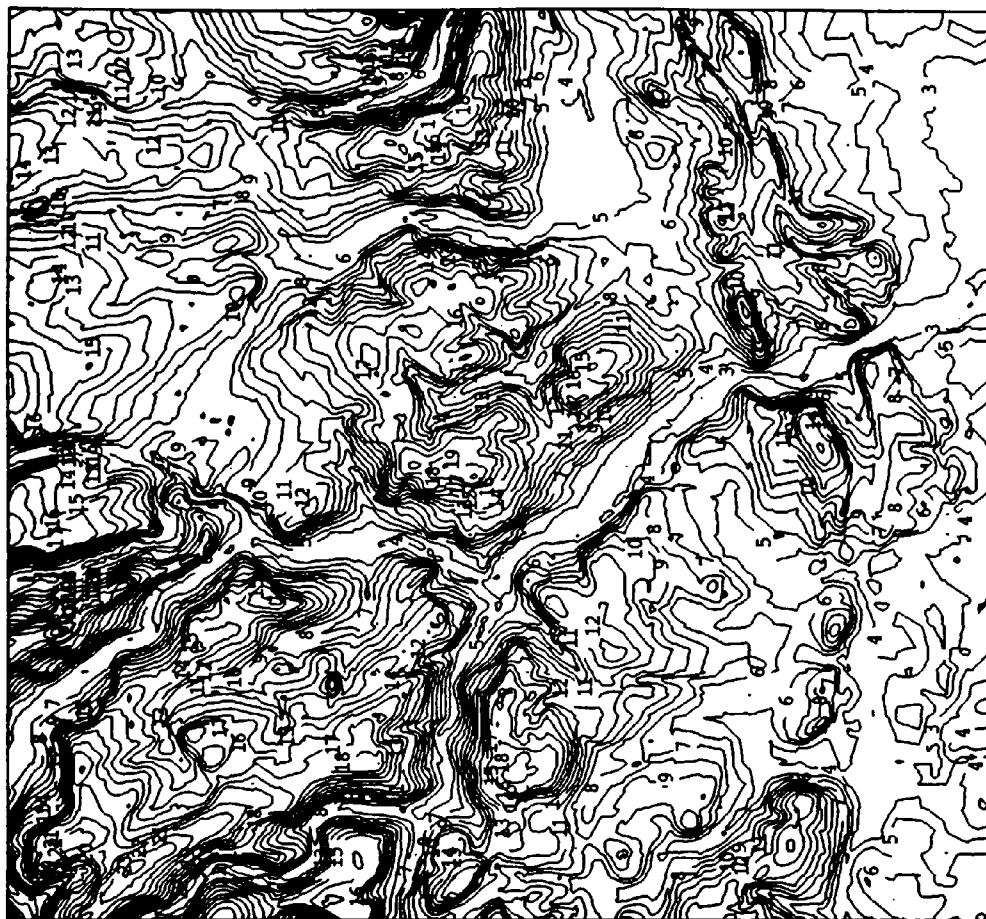


Figure 6.11 - ST08 Adaptive Polynomial Surface Patch DEM for 40x40 Patches as in Fig.6.10.

Figure 6.12 - ST08 Adaptive Polynomial Surface Patch DEM Contoured by Linear Interpolation
(Max. Abs. Error = 10 Metres, Storage = 19.293%).

Figure 6.11 and 6.12 above, illustrate the individually contoured 40x40 patches and complete surface model for the adaptive polynomial surface patch DEM illustrated in Figure 6.10, corresponding to the first entry in Table 6.8. The errors associated with this model and the performance of all the adaptive polynomial surface patches in terms of elevation are shown below in Tables 6.9 (ST06) and 6.10 (ST08).

Error Tol.	% of Grid Storage	Abs.Ave. Error	R.M.S.E.	% of Interpolated Points Within Error of				
				± 5 m	± 10 m	± 15 m	± 20 m	± 25 m
10 m	7.709	1.296	2.131	95.391	99.999	100.000	—	—
15 m	4.655	1.921	3.183	87.817	98.441	100.000	—	—
20 m	3.398	2.443	4.088	82.126	95.841	99.296	100.000	—
25 m	2.731	2.896	4.928	78.258	93.084	97.960	99.757	100.000
10 m	7.436	1.295	2.108	95.638	99.984	99.998	100.000	—
15 m	4.055	1.965	3.187	87.904	98.538	99.998	100.000	—
20 m	2.550	2.595	4.206	80.719	95.694	99.286	100.000	—
25 m	1.716	3.295	5.424	74.752	91.294	97.269	99.376	100.000
10 m	5.744	1.225	2.023	96.151	99.953	99.992	99.999	100.000
15 m	4.191	1.852	3.022	89.456	98.776	99.992	99.999	100.000
20 m	2.560	2.575	4.191	81.449	95.623	99.247	99.999	100.000
25 m	1.591	3.383	5.509	74.206	90.819	97.175	99.442	100.000

Table 6.9 - ST06 Adaptive Polynomial Surface Patch Elevation Errors.

Error Tol.	% of Grid Storage	Abs.Ave. Error	R.M.S.E.	% of Interpolated Points Within Error of				
				± 5 m	± 10 m	± 15 m	± 20 m	± 25 m
10 m	19.293	2.113	2.756	92.286	99.996	100.000	—	—
15 m	12.166	3.045	3.983	80.958	97.780	100.000	—	—
20 m	8.944	3.932	5.181	70.943	93.488	98.941	100.000	—
25 m	7.040	4.868	6.447	62.176	87.828	96.738	99.324	100.000
10 m	20.341	2.008	2.643	93.213	99.958	99.994	99.999	100.000
15 m	11.734	2.972	3.887	81.713	98.103	99.994	99.999	100.000
20 m	8.059	3.924	5.125	70.626	93.946	99.096	99.999	100.000
25 m	6.073	4.777	6.238	62.123	89.304	97.316	99.510	100.000
10 m	19.684	1.999	2.664	93.179	99.817	99.983	99.997	100.000
15 m	12.144	2.927	3.843	82.326	98.154	99.983	99.997	100.000
20 m	8.204	3.895	5.098	71.143	93.951	99.114	99.997	100.000
25 m	5.991	4.892	6.381	60.992	88.274	97.066	99.526	100.000

Table 6.10 - ST08 Adaptive Polynomial Surface Patch Elevation Errors.

These tables clearly show that whilst the 'constrained' models cannot always ensure the maximum absolute error criterion is met for some DEMs, there are only a very small percentage of points which lie outside this tolerance level. In general, this is more likely to occur for the larger surface patches (eg. 20x20 points). Further examination showed yet again that this arises due to the extreme variability within specific surface patches. For example, in the 40x40 surface patch model of ST06 illustrated in Figure 6.9, corresponding to the first entry in Table 6.9, it can be seen that the terrain along the coast line (cliffs) is modelled by 10th degree polynomials. However, despite this not all polynomially interpolated grid points are within the constrained error tolerance of ten metres. In fact, there are only two values of the original 160,000 greater than the prescribed tolerance, the largest of which being -11.811 metres. The graph of the elevation error performance in terms of root mean square error against storage is presented below in Figure 6.13, for each of the surface models of ST06 and ST08.

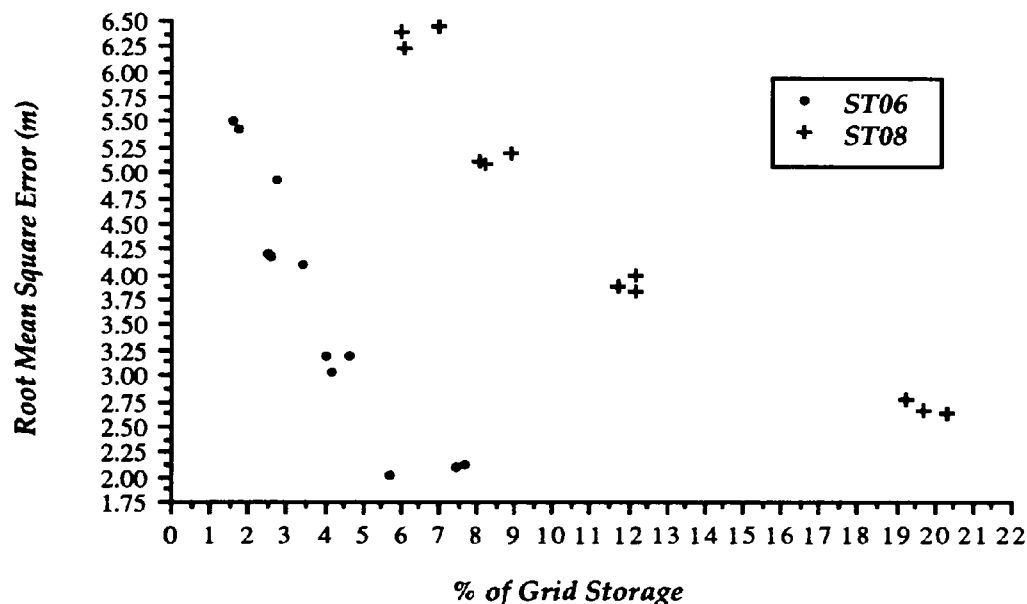


Figure 6.13 - Adaptive Polynomial Surface Patch DEM R.M.S. Elevation Errors for ST06/08.

The above figure shows a close clustering of points for each surface model of the same elevation tolerance. This suggests that the size of the surface patches has no overall effect on the performance of the models. This can be verified from Tables 6.9 and 6.10, since no model clearly stands out in terms of better elevation performance from any of other.

These results are very encouraging, since they illustrate that the random nature of polynomial functions, which were considered uncontrollable in the ordinary surface patch DEM (Section 6.3), can be closely constrained to predetermined error tolerances. The adaptive model is thus an effective surface representation technique which constrains polynomials without having to blend functions using other polynomial weighting functions. However, the continuity of the

surface along neighbouring patch boundaries is still disjoint. The general adaptive model and the effect that discontinuities might have on radio path loss was examined further. Tables 6.11 and 6.12 show the radio path loss errors associated with each of these adaptive polynomial DEMs for ST06 and ST08 respectively.

Poly. Tol. (m)	% of Grid Storage	Frequency = 200 MHz				Frequency = 400 MHz				Frequency = 600 MHz			
		A.Av. Error	RMSE	No. of Profs		A.Av. Error	RMSE	No. of Profs		A.Av. Error	RMSE	No. of Profs	
				≤ 6dB	12dB			≤ 6dB	12dB			≤ 6dB	12dB
10	7.709	1.544	2.702	960	994	1.497	2.747	942	992	1.438	2.623	947	994
15	4.655	1.798	3.092	941	991	1.871	3.464	925	985	1.820	3.362	923	989
20	3.398	1.657	2.975	942	992	1.797	3.567	920	983	1.844	3.234	923	991
25	2.731	1.323	2.672	955	993	1.472	3.375	942	985	1.461	3.121	940	991
10	7.436	1.745	2.801	954	997	1.636	2.867	934	990	1.541	2.675	946	996
15	4.055	2.180	3.419	924	988	2.154	3.708	905	983	2.045	3.501	909	988
20	2.550	2.238	3.559	909	986	2.325	3.961	896	978	2.234	3.700	898	985
25	1.716	2.260	3.626	907	984	2.363	4.133	886	970	2.350	3.963	893	979
10	5.744	1.773	2.917	943	994	1.606	2.881	940	992	1.490	2.607	948	996
15	4.191	2.398	3.786	912	986	2.278	3.826	889	981	2.254	3.737	884	988
20	2.560	2.664	4.168	884	976	2.772	4.698	855	966	2.730	4.426	858	971
25	1.591	2.714	4.267	879	973	2.912	4.789	859	960	2.896	4.629	854	966

Table 6.11a - ST06 Adaptive Polynomial Surface Patch Radio Path Loss Errors (at 200, 400 & 600 Mhz for 1000 Profiles).

Poly. Tol. (m)	% of Grid Storage	Frequency = 900 MHz				Frequency = 1400 MHz				Frequency = 1800 MHz			
		A.Av. Error	RMSE	No. of Profs		A.Av. Error	RMSE	No. of Profs		A.Av. Error	RMSE	No. of Profs	
				≤ 6dB	12dB			≤ 6dB	12dB			≤ 6dB	12dB
10	7.709	1.536	2.801	930	994	1.709	3.163	927	988	1.813	3.385	916	984
15	4.655	1.925	3.494	911	987	2.132	3.919	891	976	2.259	4.179	886	968
20	3.398	1.844	3.473	906	985	2.055	3.910	888	978	2.181	4.172	882	972
25	2.731	1.588	3.285	928	986	1.781	3.705	909	982	1.885	3.949	903	979
10	7.436	1.653	2.890	927	996	1.851	3.280	919	990	1.967	3.505	911	985
15	4.055	2.225	3.817	883	981	2.497	4.326	869	971	2.645	4.615	859	963
20	2.550	2.490	4.204	873	977	2.786	4.694	852	959	2.962	5.033	841	947
25	1.716	2.636	4.516	852	971	2.974	5.092	836	950	3.173	5.465	827	934
10	5.744	1.616	2.813	930	994	1.837	3.240	922	989	1.953	3.480	916	986
15	4.191	2.508	4.121	849	981	2.838	4.704	831	962	3.015	5.028	822	952
20	2.560	3.079	4.927	824	958	3.494	5.629	794	937	3.720	6.029	784	920
25	1.591	3.223	5.187	817	955	3.656	5.960	784	930	3.897	6.402	778	919

Table 6.11b - ST06 Adaptive Polynomial Surface Patch Radio Path Loss Errors (at 900, 1400 & 1800 Mhz for 1000 Profiles).

Poly. Tol. (m)	% of Grid Storage	Frequency = 200 MHz				Frequency = 400 MHz				Frequency = 600 MHz			
		A.Av. Error	RMSE	No. of Profs		A.Av. Error	RMSE	No. of Profs		A.Av. Error	RMSE	No. of Profs	
				≤ 6dB	12dB			≤ 6dB	12dB			≤ 6dB	12dB
10	19.293	1.977	3.406	933	981	1.813	3.194	915	988	1.663	3.095	926	986
15	12.166	2.612	4.314	884	969	2.363	4.025	881	981	2.176	3.816	896	979
20	8.944	2.872	4.884	869	965	2.784	4.722	855	964	2.669	4.626	872	959
25	7.040	3.129	5.209	848	956	3.078	5.154	838	958	2.996	5.134	846	954
10	20.341	1.863	3.331	939	984	1.708	3.105	925	991	1.520	2.934	934	991
15	11.734	2.633	4.381	888	974	2.441	4.071	873	975	2.159	3.843	889	976
20	8.059	3.006	4.847	856	967	2.943	4.810	841	963	2.565	4.305	867	968
25	6.073	3.560	5.647	814	951	3.481	5.534	803	945	3.160	5.113	834	953
10	19.684	1.953	3.421	941	980	1.756	3.186	930	989	1.585	2.952	929	992
15	12.144	2.714	4.402	885	972	2.476	4.198	878	975	2.261	4.025	884	979
20	8.204	3.330	5.182	846	950	3.069	5.033	824	958	2.847	4.797	851	956
25	5.991	3.818	5.948	800	934	3.555	5.642	801	938	3.390	5.380	810	946

Table 6.12a - ST08 Adaptive Polynomial Surface Patch Radio Path Loss Errors
(at 200, 400 & 600 Mhz for 1000 Profiles).

Poly. Tol. (m)	% of Grid Storage	Frequency = 900 MHz				Frequency = 1400 MHz				Frequency = 1800 MHz			
		A.Av. Error	RMSE	No. of Profs		A.Av. Error	RMSE	No. of Profs		A.Av. Error	RMSE	No. of Profs	
				≤ 6dB	12dB			≤ 6dB	12dB			≤ 6dB	12dB
10	19.293	1.616	3.003	927	993	1.713	3.265	921	985	1.814	3.475	914	979
15	12.166	2.197	3.892	898	975	2.347	4.263	889	969	2.487	4.569	882	961
20	8.944	2.656	4.636	870	960	2.827	5.048	871	951	2.981	5.377	860	940
25	7.040	2.970	5.075	854	956	3.168	5.537	848	946	3.334	5.893	836	935
10	20.341	1.510	2.964	936	988	1.618	3.275	928	983	1.709	3.491	923	977
15	11.734	2.170	3.875	890	977	2.349	4.310	886	969	2.489	4.605	882	960
20	8.059	2.644	4.522	861	964	2.850	4.947	854	954	3.014	5.277	841	948
25	6.073	3.246	5.363	826	950	3.541	5.936	810	936	3.750	6.334	796	924
10	19.684	1.580	3.069	922	988	1.725	3.459	916	982	1.831	3.713	911	978
15	12.144	2.256	4.007	883	981	2.443	4.446	880	964	2.585	4.751	873	949
20	8.204	2.860	4.805	849	958	3.100	5.311	846	945	3.281	5.675	833	934
25	5.991	3.413	5.494	802	945	3.659	6.007	796	931	3.863	6.400	788	918

Table 6.12b - ST08 Adaptive Polynomial Surface Patch Radio Path Loss Errors
(at 200, 400 & 600 Mhz for 1000 Profiles).

It is noticeable from the above tables, that the general overall performance of the adaptive polynomials is very good. Since the models are constrained to a great extent, in terms of maximum elevation error, the corresponding errors in radio path loss are repressed. The overall

performance of the RMS radio path loss errors are illustrated below in Figure 6.14. The number of profiles within ± 6 and ± 12 dB for each surface model are shown in Figure 6.15. Both figures average the path loss errors over 1000 profiles for all of the path frequencies (200 - 1800 MHz).

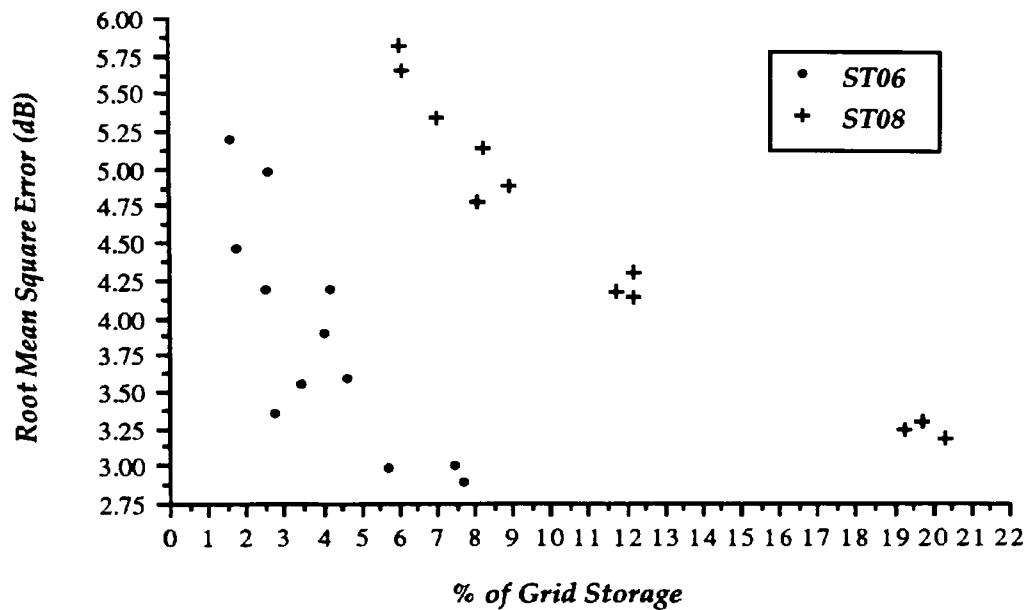


Figure 6.14 - R.M.S. Error (dBs) of Radio Path Losses for Adaptive Polynomial Surface Patch DEMs Averaged Over the Frequencies of 200, 400, 600, 900, 1400 & 1800 MHz.

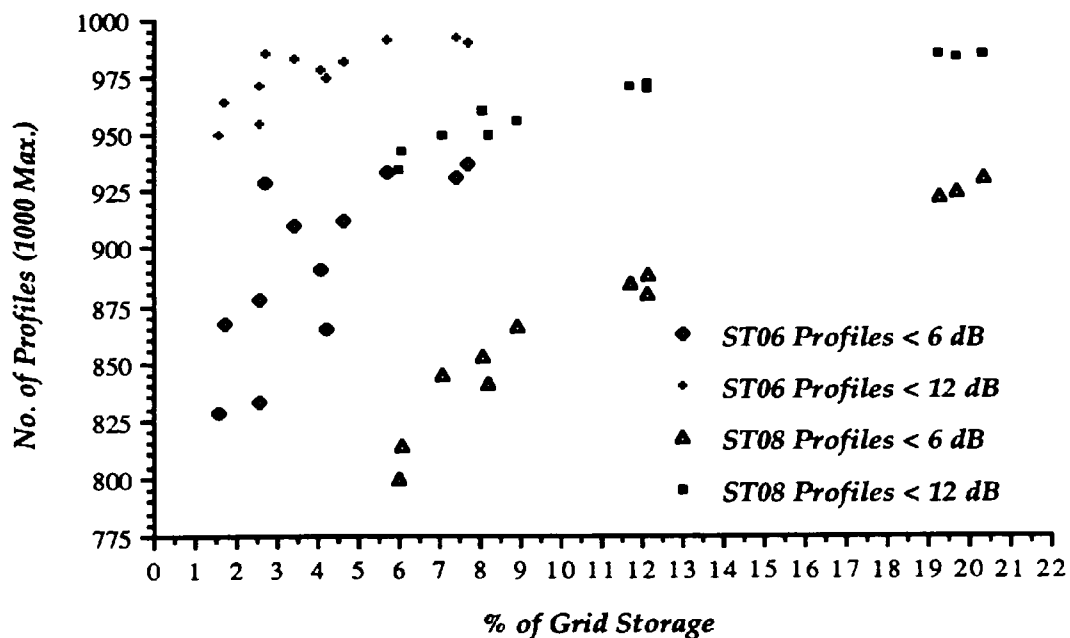


Figure 6.15 - No. of Polynomial Surface Patch Profiles Within Path Loss Error of 6 & 12 dBs Averaged Over the Frequencies of 200, 400, 600, 900, 1400 & 1800 MHz.

It is noticeable that the errors in Figure 6.14 are not as clustered as the RMS elevation errors in Figure 6.13, particularly for ST06. These more variable results are not too significant, since

these generally occur in the surface models where storage savings in excess of 95% of the original grid are attained. As storage increases, and the elevation models become more constrained, the path loss errors behave more consistently for equivalent adaptive polynomial DEMs. These characteristics are also reflected in Figure 6.15, for the number of profiles within ± 6 and ± 12 dBs.

Figures 6.14 and 6.15 both conclusively show that adaptive polynomial surface patch modelling produces better path loss results than for the unconstrained polynomial DEMs (in Figures 6.7 and 6.8), for equivalent levels of storage. This is true for all the models tested, since no DEM of equivalent storage in Section 6.3 produces better results in terms of elevation or radio path loss errors.

Hence, it can be concluded that polynomial modelling can provide a viable alternative to the regular grid DEM, provided that some constraints on elevation error can be incorporated. This has been shown to be accomplished with the adaptive polynomial surface patch DEM. Variation of the polynomial order within each surface patch allows a degree of independence, which is a necessity if the model is to adapt to the terrain variability. This allows significant storage savings to be attained, without significant elevation or radio path loss errors. The flexibility of this approach and its adaptability allows polynomials consisting of between one and 66 coefficients to represent surface patches of between 100 and 400 original grid points. Storage savings in excess of 80% can be attained for ST08 (92% for ST06), whilst limiting absolute error to ten metres, in general. This results in radio path loss errors of less than 3.25 dBs (root mean square error), for both ST06 and ST08 averaged over all tested DEMs and frequencies for 1000 profiles. This corresponds to 92.6% and 98.5% of profiles within ± 6 and ± 12 dB absolute error for ST08 (93.4% and 99.2% for ST06).

It is possible that the storage of the four-byte polynomial coefficients may be reduced by data compression. However, there is no simple, error-free method which efficiently achieves this. Jancaitis (1978) considers the magnitude of low-degree polynomial coefficients, such that if the integer part is bounded by ± 128 , the coefficient is represented by 8 bits. Further compression of the coefficients may be possible if they are difference-encoded. Tests carried out on the polynomial functions discussed in this chapter suggest that such an approach is only feasible if the number of coefficients is less than six. The importance of some coefficients is critical, such that generalisation or 'rounding' of values creates additional elevation errors. It was concluded therefore, that the storage savings generated from compressing coefficients was not worth the additional errors and processing time required to analyse coefficients and assign scaling constants.

Chapter Seven

Surface Patch Quadtree

7.1 Introduction

The advantages of representing DEMs by rectangular grid cell methods have been outlined extensively in the preceding chapters. These not only include the direct representation of elevations at grid vertices, but also mathematical DEMs, spatially addressed with a regular grid. It was shown in Chapter Six that polynomial surface patches can offer substantial storage savings over the regular grid, but elevation errors are unconstrained. This is due to the fixed-degree polynomials approximating the general trend of a surface, but being unadaptable to the local fluctuations of variable terrain. This problem was partly overcome by varying the degree of the polynomial within each fixed-size surface patch. However, an alternative approach is to vary the size of the surface patch or grid, whilst using an approximating function of fixed degree. This can be achieved with the use of a hierarchical data structure, in which homogeneous regions (ie. areas of uniform terrain) are referenced by a single key identifier and corresponding surface function. One such data structure based on a square tessellation is the quadtree. More specifically, a quadtree in which terrain data can be partitioned and represented by approximating functions is termed the surface patch quadtree.

The quadtree can be viewed as a variant of the regular grid, but allows high compression of the number of grid cells required to represent an image (or surface) in uniform regions. This is accomplished by retaining much of the simplicity associated with grid representations (Abel, 1985). It can be compared to the variable density grid, discussed in Chapter Four. Samet (1984) provides an overview of hierarchical representations and has recently produced a more comprehensive examination of such data structures and their applications (Samet, 1990a; 1990b). These reviews consider the various forms of quadtree in great detail. A significant advantage of the quadtree over other hierarchical tessellations such as a triangulation is that the grid can be recursively subdivided with areas of both the same shape and orientation.

Quadtrees represent two-dimensional (spatial) data in a way which takes advantage of spatial coherence in the phenomena being represented (Mark, 1986; Mark et al, 1989). As such, they have received considerable attention as a data structure for image processing and GIS applications (Cebrian et al, 1985; Mark, 1986), including approaches for handling diverse spatial data types; strategies for covering very large areas; and applications to problems in computational geometry, spatial search and spatial modelling.

Quadtrees are used to recursively decompose a regular square ($2^n \times 2^n$) image, picture or surface

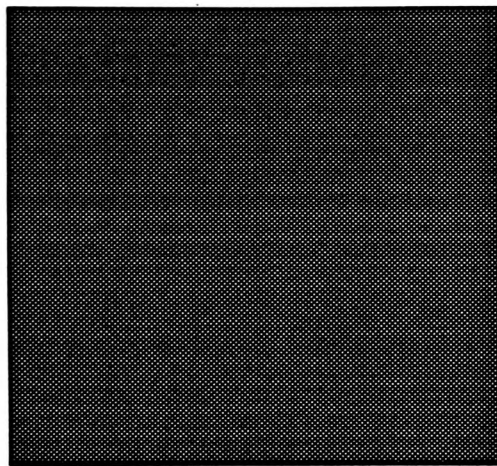
into smaller, homogeneous square areas using a distinct sub-division criterion. This is repeated for each quadrant until no further sub-division is possible and each sub-quadrant is homogeneous. The quadtree data structure describes the successive partitions that achieve this. It is most commonly used to separate an object or region from its background image. For example, consider the region quadtree of the image in Figure 7.1, overleaf.

The object or region (ie. the black pixels in Figure 7.1d) is first considered as a single node of 8x8 pixels (Figure 7.1a). Since the image needs further decomposition, this grey node is sub-divided into four quadrants of 4x4 pixels (Figure 7.1b). All grey nodes that are intersected by the object (ie. non-homogeneous) are recursively sub-divided until all nodes lie within the object (black or leaf nodes) or outside (white nodes). The tree representation corresponding to this decomposition is shown in Figure 7.1e.

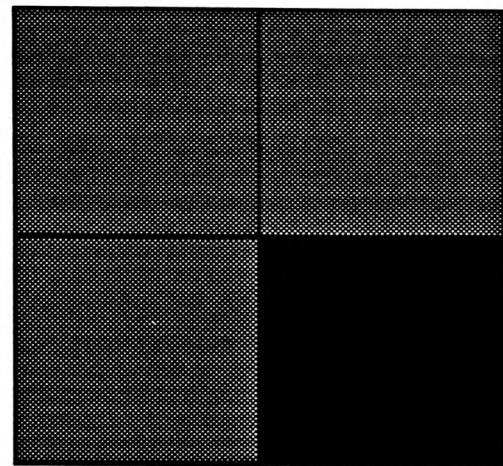
Early work on quadtrees represented relations among nodes by an explicit tree structure, with nodes linked to parents and children by pointers (Samet, 1984). Algorithms have been described in the literature which require elemental operations for tree traversal and for examination of neighbouring nodes, with the space required to store a quadtree and solution times to execute an operation dependent on the structure adopted to represent the quadtree itself. However, the pointers in an explicit quadtree are too costly in terms of storage for an efficient digital terrain model.

Quadtrees appear to have many advantages for handling coherent 'blocky' spatial data, but are inefficient for continuous surfaces such as topography (Mark, 1986). However, a number of strategies have been developed in order to model terrain data, since the benefits of the quadtree have been recognised as a major contribution to the integration of data within GIS. This emphasis is not only on the integration of elevation data, but also on strategies for handling coverage files (Mark et al, 1989), such as details of land use, vegetation or man-made objects. The incorporation of such data sets could be of benefit for a terrain modelling system for calculating radio path losses, for example.

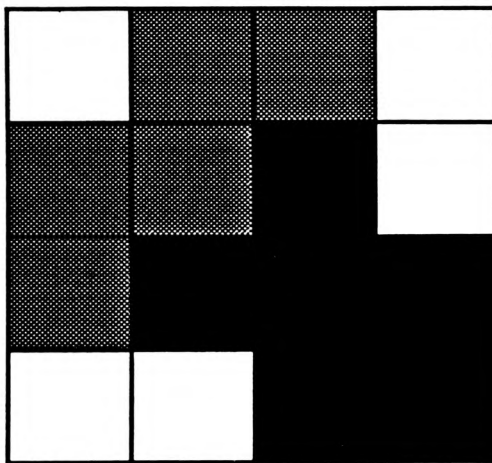
Martin (1982) describes a database system for geographical data based on quadtrees. Descriptors are used to define the geometrical properties of surface data using equations that offer more global information and efficient geometrical analysis. Samet et al (1983) provide a quantitative assessment of the efficiency of quadtrees as a means of representing regions in a cartographic database, which includes terrain elevations, land use classes and floodplain boundaries. The surface patch quadtrees of Chen & Tobler (1986) and Leifer & Mark (1987) attempt to combine the advantages of mathematical functions within a local surface patch network. The inflexibility of some of some functions in adapting to the terrain variability is overcome by varying the region of validity or patch for each surface function.



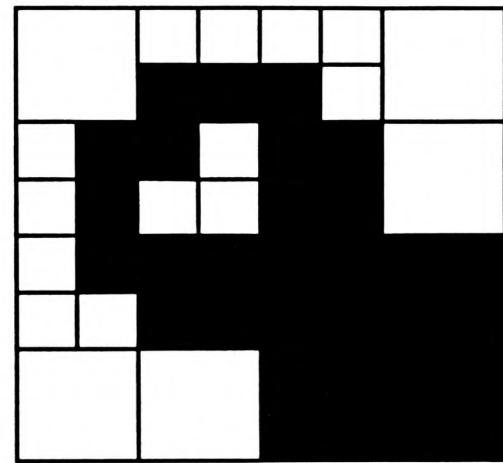
(a) Level 0



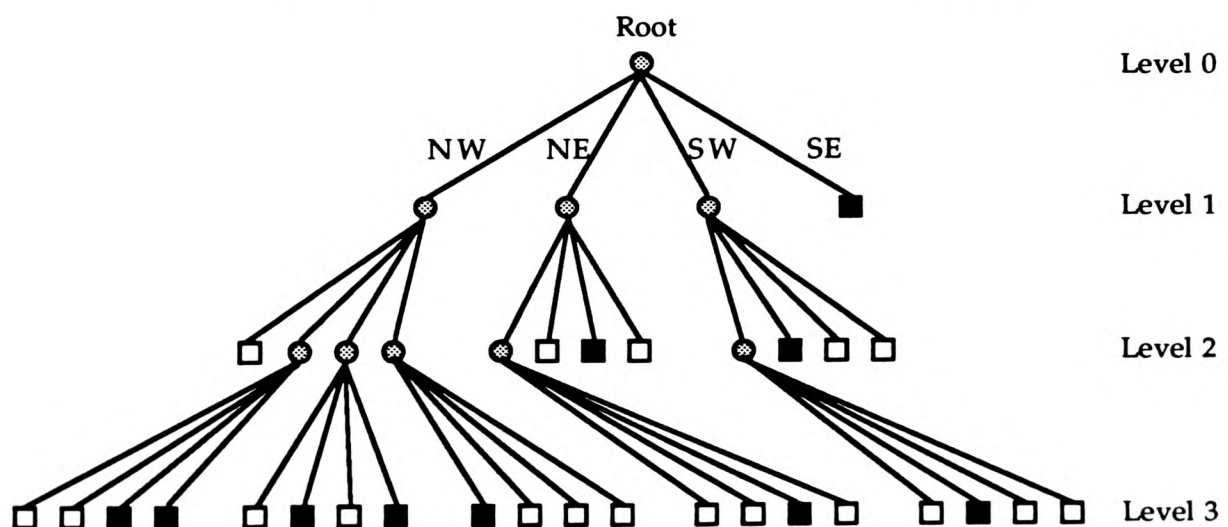
(b) Level 1



(c) Level 2



(d) Level 3



(e) Quadtree Representation

Figure 7.1 - Decomposition of a Region into its Quadtree Representation.

7.2 Linear Quadtree

One class of space-efficient quadtree, termed the linear quadtree (Gargantini, 1982) represents only leaf nodes, identified by numeric keys. The form of these keys permits topological and spatial relations to be determined from the key values through bit manipulations or modular arithmetic (Mark, 1986). The data structure is thus a list of leaf nodes, in sequence by key. However, geographic data are essentially two-dimensional, whereas computer storage and processing are essentially one-dimensional (Mark, 1986). As such, no linear sequence can preserve all the spatial properties of geographic data.

This problem of preserving the spatial properties is more significant with the use of compressed data structures, such as the linear quadtree. The main advantages of the linear quadtree with respect to conventional quadtrees are that the pointers are eliminated, and the space and time complexity depend only on the number of leaf nodes. Dutton (1983) highlights the fact that the storage of linear quadtrees is about twice as compact as non-linear ones. However, the price to be paid for this may be significant increases in processing time. Each field of the linear quadtree must be examined sequentially to compute the locations of the encoded nodes, whereas with pointers, random access to quadrants can be considerably faster (Dutton, 1983).

Two of the most popular forms of fixed-length keys for linear quadtrees are presented by Gargantini (1982) and Abel & Smith (1983). The key indexing of the former quadtree is illustrated below in Figure 7.2a.

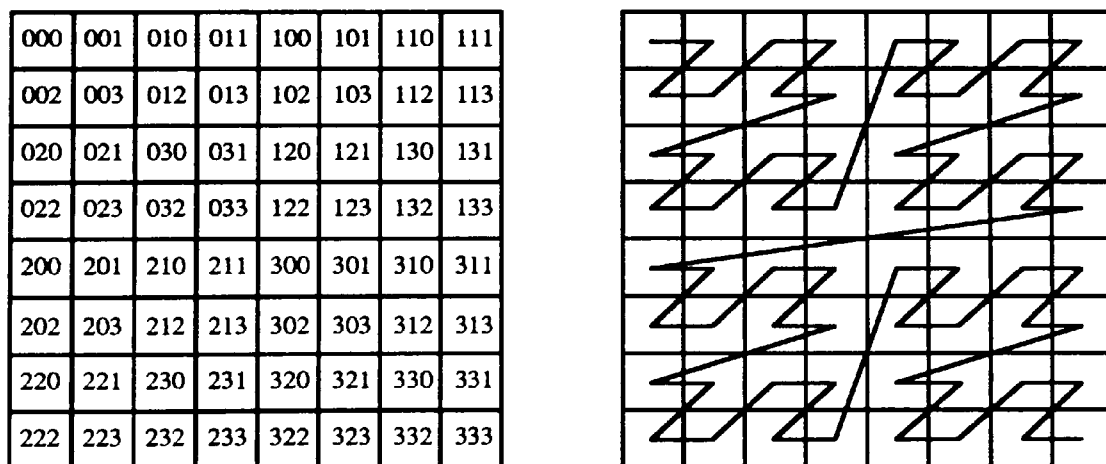


Figure 7.2 - (a) Linear Quadtree Labelling Scheme and (b) Sequential Search Order.

Each node is encoded as a weighted quaternary code (with digits 0, 1, 2, and 3 in base 4), where each successive digit represents the quadrant subdivision from which it originates (Gargantini, 1982). The NW quadrant is encoded with 0, the NE with 1, the SW with 2 and the SE with 3. The linear search for a node would follow the sequence illustrated in Figure 7.2.b However,

within the linear quadtree, only leaf nodes are stored, so there is no data redundancy. The compression of nodes is allowed, such that four leaves within the same quadrant are replaced by a higher level node, signified by a code with a marker in the last digit. For example, consider the object in Figure 7.3, below.

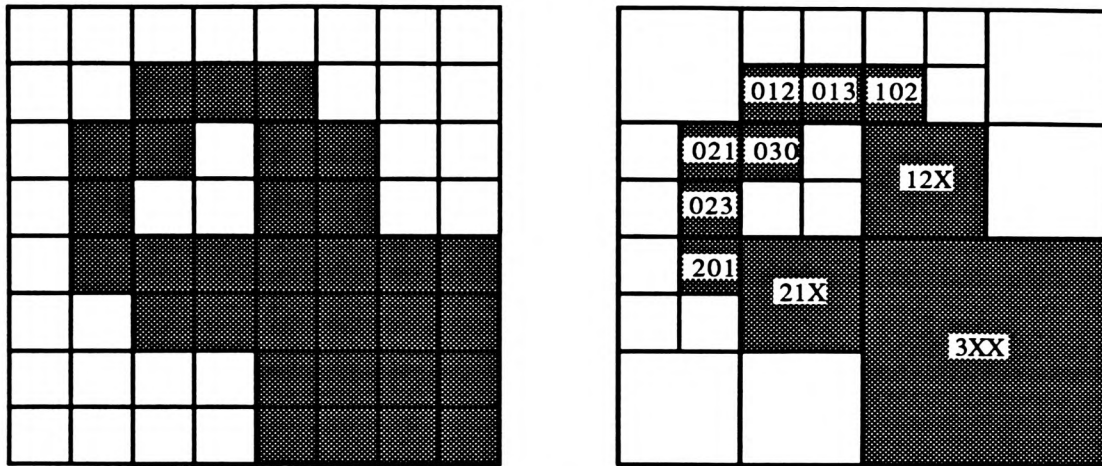


Figure 7.3 - Representation of an Object within the Linear Quadtree at its Lowest and Highest (Compressed or Condensed) Levels.

This object can be represented by the nodes 012, 013, 021, 023, 030, 102, 12X, 201, 21X and 3XX, where X is a marker indicating a condensation, or a node at a higher level of the quadtree. For example, 12X signifies a condensation of the four nodes 120, 121, 122 and 123. This ordered list of nodes follows the search pattern illustrated in Figure 7.2b. A suitable value for the marker (X) is an integer greater than three.

The linear quadtree is a storage-efficient data structure for representing regions, since only the black nodes are stored and coordinate information can be obtained by decoding the key values. For terrain modelling however, the quadtree will contain no white nodes (ie. empty or redundant quadrants), since the complete surface will be modelled.

For this research, the form of the key is critical for storage to be minimised. A two-byte key was required which references all the nodes in a 256 x 256 pixel surface, where the size of the surface patches varies between 128x128 pixels (Level 1) and 2x2 pixels (Level 7). It was necessary for the key to be represented in base five, (0-3 for the four quadrants, and a marker of 4 to indicate a higher level). The range of quadtree key is therefore 0000000 to 3444444 (in base 5) or 0 to 62,499 as a decimal, which can be represented in sixteen bits. When constructing the linear quadtree, the surface is first considered as an initial four quadrants, rather than one root node, since there will always be a need for the surface to be sub-divided. Hence, level 0 of the DEM quadtree (ie. the 256x256 surface) is implicit and does not require a key. The initial DEM quadtree and its first level nodes are represented below in Figure 7.4.

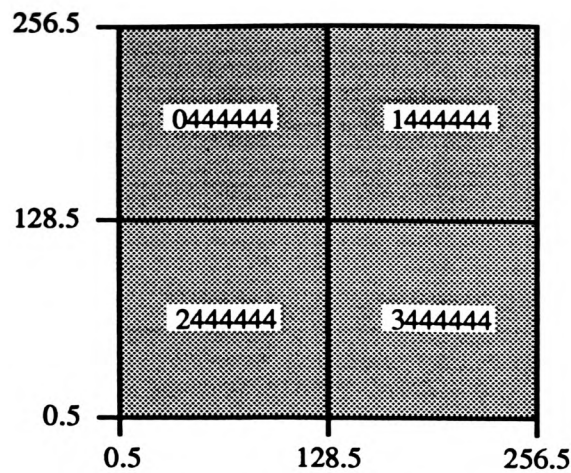


Figure 7.4 - Initial Quadtree Representation (Level 1).

During construction of the quadtree, the key of a node to be sub-divided is replaced by four new keys (nodes). These take the form of the parent node up to the most significant marker, which is then replaced by the code for each son (ie. 0, 1, 2 and 3). The coordinates of any node and its size can be calculated from the two-byte integer key. By initially setting each node's coordinates to 0.5, 0.5 (ie. the SW corner of the quadtree), each key digit is read in turn, such that its position in the quadtree can be calculated. The algorithm for decoding the quadtree key is presented below in Figure 7.5 and an example of its use is shown in Figure 7.6.

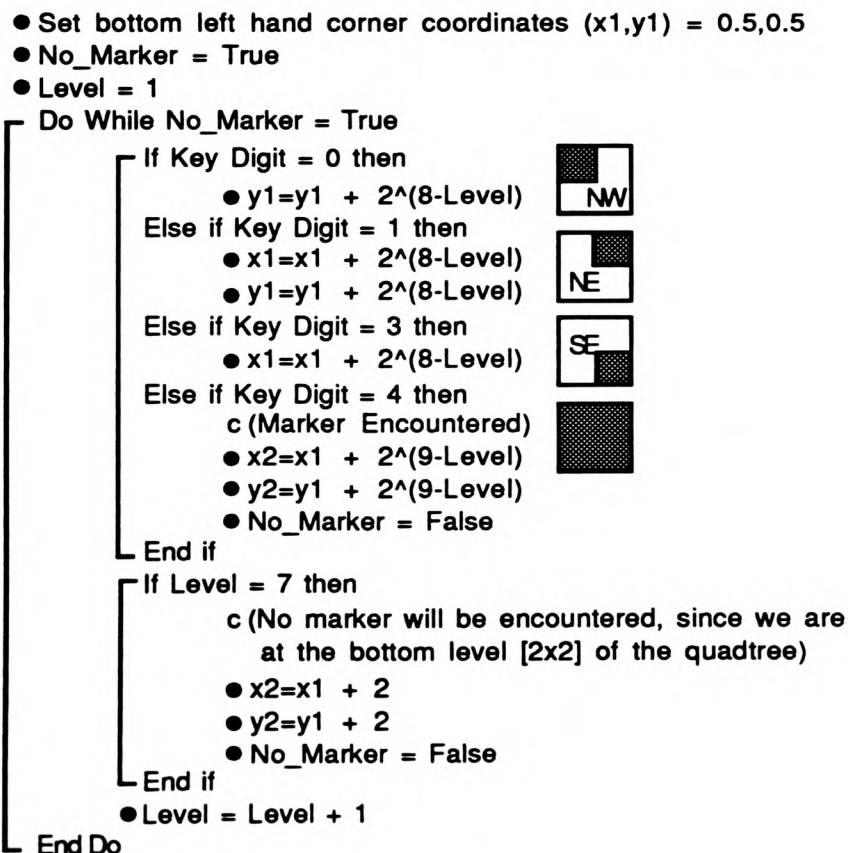


Figure 7.5 - Algorithm for Decoding a Quadtree Key. (N.B. No adjustment if Key digit = 2). (x1,y1 and x2,y2 are the bottom left and top right coordinates respectively).

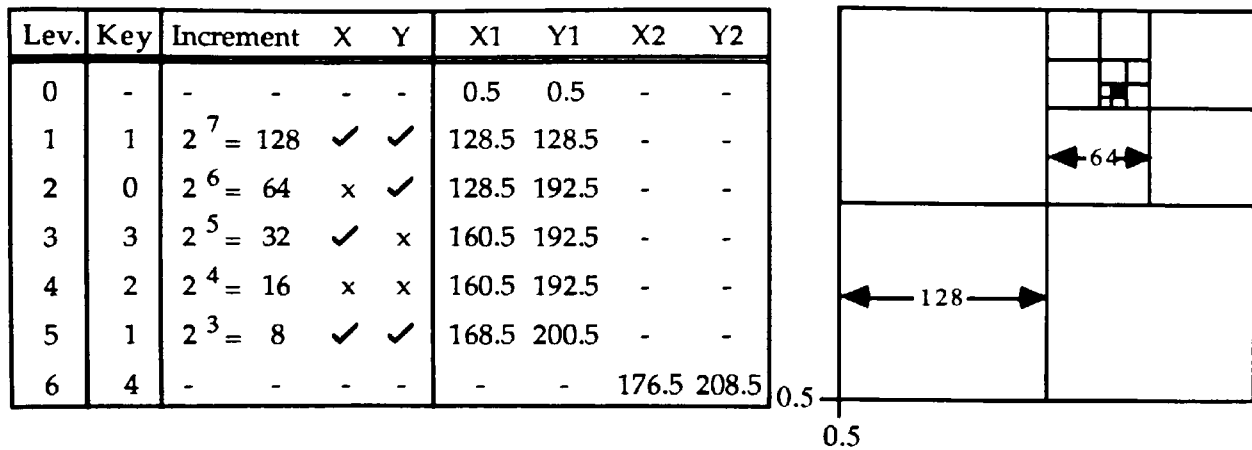


Figure 7.6 - Illustration of Key Decoding Algorithm (Key = 1032144).
(Therefore Node is an 8x8 Patch Whose SW Coordinates are 168.5, 200.5).

7.3 Mathematical Surface Patch Functions

The surface patch quadtree of the regular grid DEM is constructed by applying a mathematical function to the vertices of each quadtree node. The size of these nodes or patches is dependent upon the nature of the terrain and the flexibility of the approximating function. However, the length of each quadtree node (ie. pixels per square edge) will be an exact power of two (ie. 2^n , where $n = 1$ to 7). As shown for the polynomial surface patches in Chapter Six (Figure 6.3), a more storage efficient DEM is obtained by assuming that the grid elevations are at the centre of each grid cell, rather than at the vertices.

The actual grid elevation $z(i,j)$ is approximated by a normalised local surface function $f(x,y)$ satisfying the equation:

$$\text{diff}(i,j) = |z(i,j) - f(x,y)| < \text{tol} \quad \dots [7.1]$$

where **tol** is the absolute maximum allowable error. The value of **diff** is calculated at each grid elevation, such that if any point is outside the tolerance, the surface patch node is sub-divided.

The form of the mathematical function used to approximate the terrain was given considerable attention. Least squares polynomials were rejected because of their expensive four-byte coefficients and the need for a simple function that can approximate patches of 128x128 vertices to 2x2 elevations. Leifer & Mark (1987) use orthogonal polynomials in their representation of the surface patch quadtree, but constraints are imposed on the size of the patches. Large patches create computational problems, whilst small patches offer little or no saving in storage costs. However, Leifer & Mark's quadtree is constructed in terms of R.M.S. error for each patch. The additional constraint of defining a maximum error tolerance would most probably increase the storage costs of this DEM.

Chen & Tobler (1986) consider five simple mathematical functions for approximating the surface plane within each quadtree node:

- (a) The average elevation of all pixel heights;
- (b) The maximum elevation of all pixel heights;
- (c) The minimum elevation of all pixel heights;
- (d) The equation of a ruled surface (hyperbolic paraboloid) or bilinear interpolant:

$$f(x,y) = a + bx + cy + dxy \quad \dots [7.2]$$

- (e) The equation of a quadric surface:

$$f(x,y) = a + bx^2 + cy^2 + dxy \quad \dots [7.3]$$

The coefficients of equations [7.2] and [7.3] are formed from the four corner elevations of the quadtree node (for a more detailed description, see Chapter Three, p.51). Hence, for these surfaces, the four integer elevations need to be stored with the quadtree key. This requires a total of ten bytes of storage per node. However, if the elevations are difference encoded, only four bytes are needed for the four elevations and two bytes for the key. For the average, maximum and minimum surfaces, only one elevation value needs to be stored with the key, requiring four bytes of storage per node.

Daly (1989) suggests another approach which uses the four corner points of each quadtree node. However, accuracy is measured using the four triangles which the corner points make with an averaged centre point. Linear interpolation through these triangles ensures continuity at the edges. Tesser and DeMund (1989) construct a surface patch quadtree (or Laplacian pyramid) by computing the difference between two Gaussian pyramids (one smoothed and sub-sampled whilst the other is just smoothed). If a threshold is exceeded (due to variable terrain) the region is sub-divided in a recursive top-down approach.

Experimental investigations were carried out for each of these approaches and for a variety of criteria used to subdivide the quadtrees (ie. maximum error, average error, RMS error etc.). However, the simplest approach (and most storage efficient) is that of Chen & Tobler (1986) with quadtree subdivision accomplished on the basis of maximum absolute error. Each of Chen & Tobler's surface functions were tested in a prototype implementation of a surface patch quadtree for 256x256 subsets of the data sets ST06 and ST08. Quadtrees were constructed at different tolerance levels, such that the storage costs of each surface could be analysed. Despite the limited storage requirements of the average, maximum and minimum surfaces, the most consistent surface function is the ruled or bilinear surface. The greater flexibility of this function does not create the excessive node sub-divisions of the level plane surfaces. The quadric surface was found to be too inconsistent for terrain approximation. Therefore, for all the

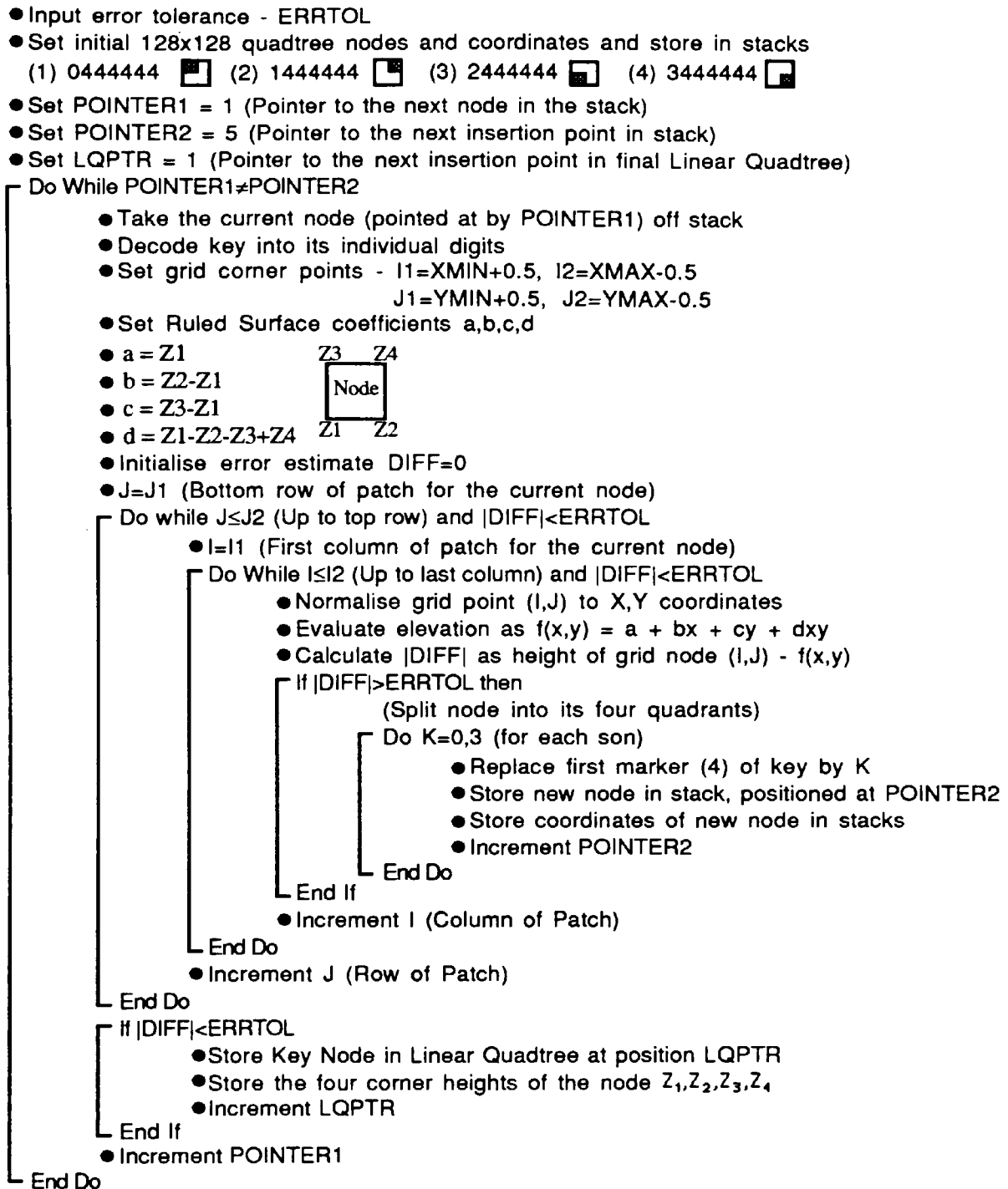
surface functions considered and examined, the ruled surface was chosen as the local function which best approximates terrain, whilst being simple and storage efficient. This conclusion is consistent with the results of Chen and Tobler's research (1986). Jancaitis (1978) also suggests the use of this surface function for elevation data compression.

7.4 Surface Patch Quadtree Implementation

The task of fitting a surface patch quadtree to a regular grid DEM consists of defining mathematical functions for each node in the quadtree. The original grid values are re-interpolated from each surface function, so that if a point is not within some preset error tolerance, the node will be sub-divided. As the patches (nodes) become smaller in area, the mathematical functions get progressively better in terms of goodness of fit. Hence, the final quadtree will represent a fine balance between the flexibility of the bilinear surface function and the variability of the terrain.

Each key of the initial quadtree (Figure 7.4) is placed on a stack representing the nodes which are to be modelled (Figure 7.7a). The coordinates of these nodes may also be stored. The key for each node is removed from the stack in turn, and the bilinear surface is fitted to the four corner elevations. If the surface function is not a good enough approximation to the original grid nodes, the keys for the four sons of the node are added to the stack. If the error-criterion is satisfied, the key of the current node is placed in the linear quadtree, together with the elevations at its four corners (which form the coefficients of the bilinear surface function). This process is repeated until there are no more nodes in the stack and the surface has been completely defined. The algorithm for generating this quadtree is presented overleaf in Figure 7.7.

A limitation of the surface patch quadtree is the constrained size of the DEM. For the O.S. 401x401 pixel data sets, the largest derived quadtree is 256x256 pixels. Re-interpolation of the O.S. data sets to 256x256 or 512x512 pixels is not feasible, since important features may be lost or too much redundant data added. Therefore, the surface patch quadtree was applied to subsets of the original 401x401 data sets of ST06 and ST08. Four overlapping 256x256 subsets were used as the basis for forming the surface patch quadtrees at various error tolerances. These are illustrated in Figures 7.8 (for ST06) and 7.9 (ST08). The storage requirements of these quadtrees are presented as a percentage of the original grid storage for ST06 and ST08 (Tables 7.1 and 7.2). This assumes that each quadtree node requires six bytes of storage, compared to the original two-byte grid elevations.



	Node	Xmin	Xmax	Ymin	Ymax
Ptr1	0444444	0.5	128.5	128.5	256.5
	1444444	128.5	256.5	128.5	256.5
	2444444	0.5	128.5	0.5	128.5
	3444444	128.5	256.5	0.5	128.5
Ptr2					

(a)

	Node	Xmin	Xmax	Ymin	Ymax
Ptr1	1444444	128.5	256.5	128.5	256.5
	2444444	0.5	128.5	0.5	128.5
	3444444	128.5	256.5	0.5	128.5
	0044444	0.5	64.5	192.5	256.5
Ptr2	0144444	64.5	128.5	192.5	256.5
	0244444	0.5	64.5	128.5	192.5
	0344444	64.5	128.5	128.5	192.5

(b)

Figure 7.7 - Algorithm for Creating the Surface Patch Quadtree and Illustration of
 (a) Initial Stack and (b) Stack after Subdivision of the First Node (0444444).

ST06 Surface	% Storage Costs within Error Tolerance of				
	± 25m	± 20m	± 15m	± 10m	± 5m
1	1.27	1.97	3.18	5.92	11.79
2	0.99	1.60	2.79	4.84	9.92
3	1.65	2.76	4.55	8.53	20.04
4	1.68	2.59	4.23	7.72	16.61
Ave.	1.40	2.23	3.69	6.75	14.59

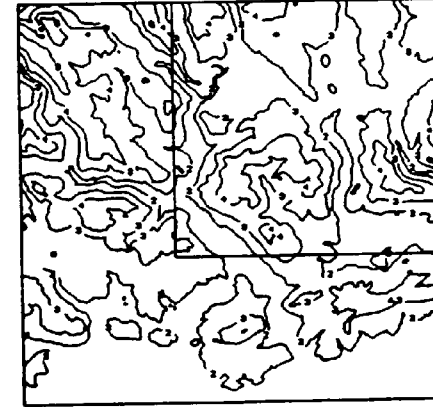
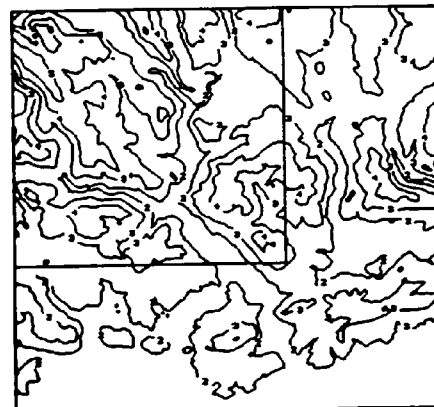
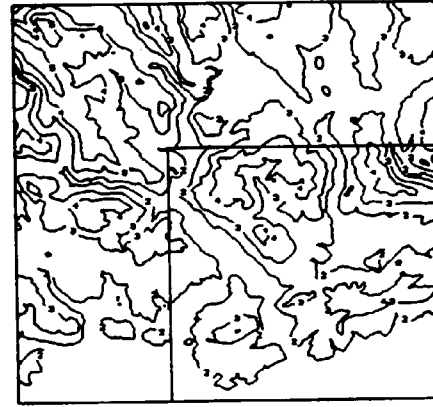
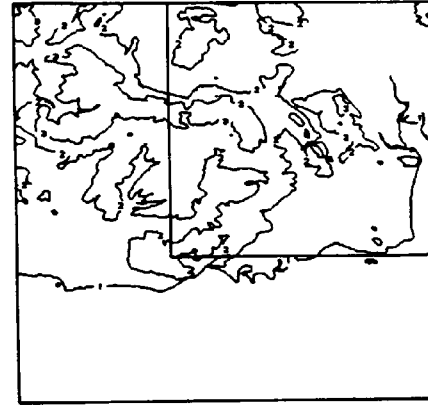
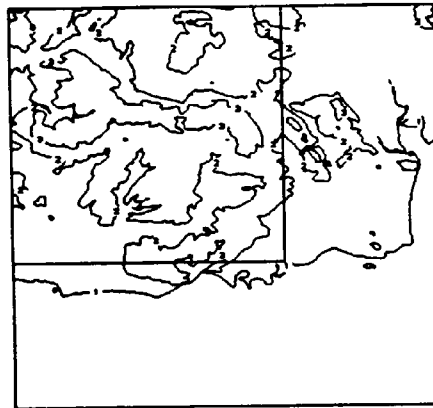
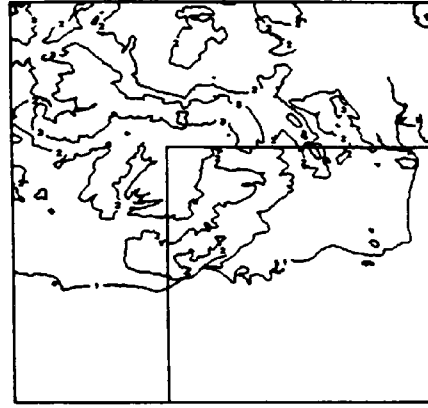
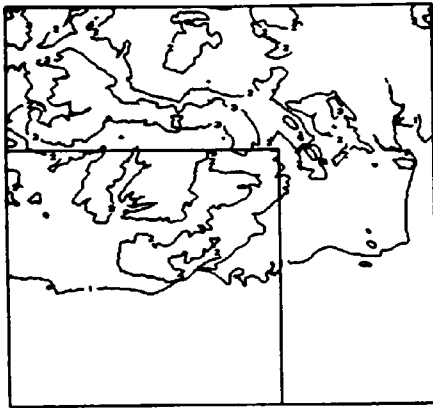
Table 7.1 - % Storage Costs for the Surface Patch Quadtrees of the Four Subsets of ST06.

ST08 Surface	% Storage Costs within Error Tolerance of				
	± 25m	± 20m	± 15m	± 10m	± 5m
1	5.42	7.56	10.57	16.85	34.84
2	5.44	7.19	10.35	16.46	32.95
3	7.27	9.66	13.72	21.37	41.59
4	5.83	7.65	10.85	17.09	34.12
Ave.	5.99	8.01	11.37	17.94	35.87

Table 7.2 - % Storage Costs for the Surface Patch Quadtrees of the Four Subsets of ST08.

From Tables 7.1 and 7.2 it can be seen that there are noticeable differences in storage between subsets of the same data sets, despite overlapping surface models (for example, subsets 2 and 3 for both ST06 and ST08). This not only illustrates the distinct changes in the nature of terrain within any surface, but also the adaptable nature of the surface patch quadtree in modelling this variability. In particular, the extensive storage savings for subsets 1 and 2 of ST06 (Table 7.1) can be attributed to the large proportion of sea level elevations in the original regular grid DEM (Figure 7.8, parts 1 and 2). For example, in this second surface there is a 6-byte SE node at the first level of the quadtree representing 128x128 2-byte elevations at sea level.

Figure 7.10 illustrates the adaptability of the surface patch quadtree for a subset of ST06 at an absolute maximum error tolerance of 5 metres. This region (part of the South Wales coastline) was selected since it highlights the flexibility of the quadtree for terrain of differing variability. It can be clearly seen that the largest sub-division of quadtree nodes occur in critical regions of the surface, such as steep slopes or at a concentration of contours. In some of these instances the quadtree is at its deepest level (ie. 2x2 grid nodes or pixels), such that the four elevations of these nodes represent the terrain at its original 50 metre grid sampling density. Whilst the storage savings in these regions are minimal, the greatest compression occurs for the large areas of uniform terrain, such as at sea level. In Figure 7.10, the terrain which is represented by quadtree nodes of 16x16, 32x32 and 64x64 pixels covers over 28.5% of the surface, but can be stored in under 0.5% of the original grid DEM requirements.



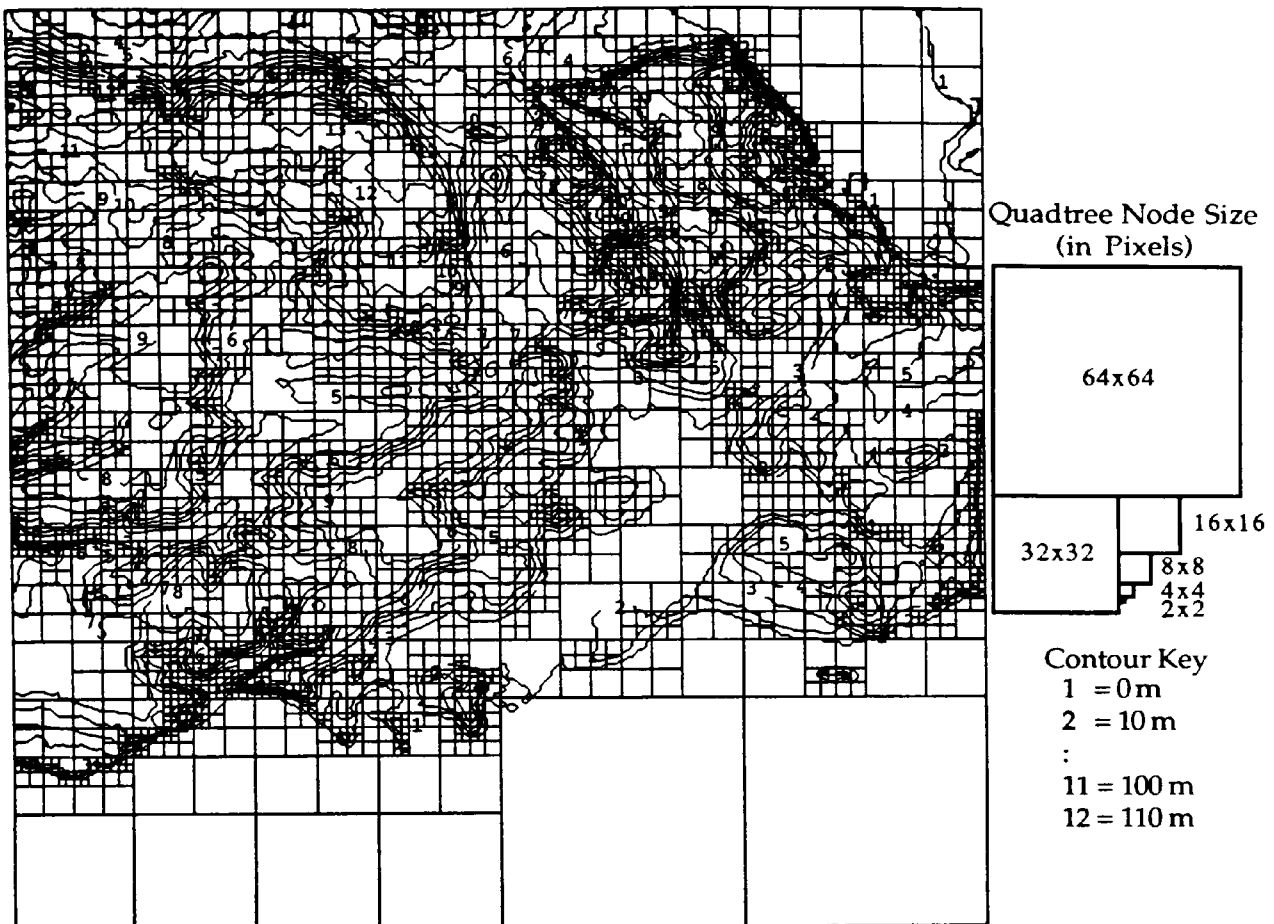
(1) X = 1 to 256
Y = 1 to 256

(2) X = 146 to 401
Y = 1 to 256

(3) X = 1 to 256
Y = 146 to 401

(4) X = 146 to 401
Y = 146 to 401

Figures 7.8 and 7.9 - Illustration of the Four 256x256 Subsets of the Original 401x401 Data Sets for ST06 and ST08.



*Figures 7.10 - Surface Patch Quadtrees for a Subset of ST06 (120,70 - 375,325)
at Maximum Error Tolerances of ± 5 metres (3544 Nodes).*

Since the surface patch quadtrees are applied to subsets of the original data sets ST06 and ST08, the results are not directly comparable with the other digital elevation models presented within this thesis. However, in order to make some form of comparison, the results for each data set are averaged over the four subset quadtrees in Figures 7.8 and 7.9. These averaged storage costs for ST06 and ST08 are presented in Tables 7.1 and 7.2. All future results presented in this chapter are averaged over the four surface patch quadtrees for each data set. Therefore, there may be a small margin of error associated with each result. However, since the subsets tend to overlap in regions of variable terrain, the storage costs and elevation errors may be over-exaggerated (for example, all four subsets of ST08 in Figure 7.9 include a section of the Taff valley). The general results in terms of elevation errors for the surface patch quadtree are presented overleaf in Tables 7.3 (for ST06) and 7.4 (ST08). Furthermore, the relationship between root mean square error and storage is illustrated in Figure 7.11.

Max.Error Tolerance	% Storage	Abs.Ave. Error (m)	RMSE	Stan. Dev.	% of Points within		
					± 5 m	± 10 m	± 15 m
5m	14.59	0.62	1.09	1.25	100.00	-	-
10m	6.75	1.43	2.33	2.71	93.39	100.00	-
15m	3.69	2.29	3.59	4.17	83.69	97.67	100.00
20m	2.23	3.23	4.99	5.79	74.40	92.28	98.76
25m	1.40	4.22	6.44	7.58	66.73	86.17	95.41

Table 7.3 - ST06 Surface Patch Quadtree Elevation Errors.

Max.Error Tolerance	% Storage	Abs.Ave. Error (m)	RMSE	Stan. Dev.	% of Points within		
					± 5 m	± 10 m	± 15 m
5m	35.87	0.68	1.22	1.41	100.00	-	-
10m	17.94	1.81	2.71	3.25	91.22	100.00	-
15m	11.37	2.88	4.12	5.01	79.36	96.88	100.00
20m	8.01	4.02	5.59	6.84	68.33	90.71	98.46
25m	5.99	5.22	7.13	8.77	58.92	83.54	94.78

Table 7.4 - ST08 Surface Patch Quadtree Elevation Errors.

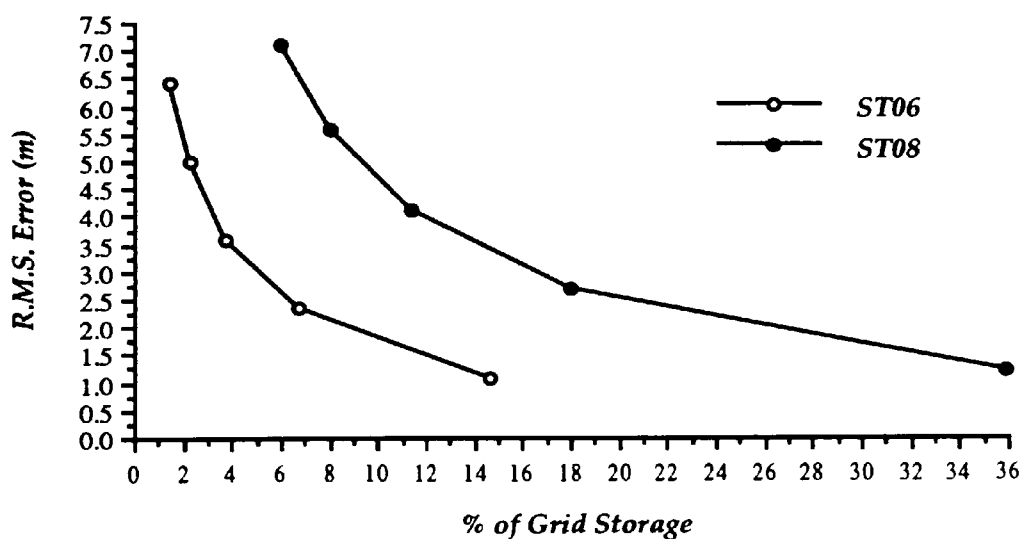


Figure 7.11 - Root Mean Square Errors of Surface Patch Quadtrees.

The surface patch quadtree is adaptable to the terrain features of ST06 and ST08, irrespective of the range in elevation. However, as the terrain becomes more variable, the storage costs of the quadtree increase, particularly as the error tolerance decreases. In comparison to the variable degree polynomial surface patch DEMs of Chapter Six, the surface patch quadtree does not offer any significant storage saving for similar elevation errors. The variable-size patch and fixed-degree function of the quadtree contrast sharply with the fixed-size patch and

variable degree function of the adaptive polynomial DEM. However, in terms of elevation error and storage, the relative merits of both approaches appear to be in equilibrium.

7.5 Profile Interpolation and Radio Path Loss Performance

The ideal method of profile interpolation within the quadtree is to follow the path of the profile from node to node. More specifically, such an algorithm requires finding the node adjacent to a given one in a specified direction. Gargantini (1982) discusses some of the issues in determining the adjacent node of a pixel at the lowest level. However, for the more general case of this quadtree, four possible adjacency instances can be distinguished in any direction:

- (i) Neighbouring node is in the same quadrant at the same level;
- (ii) Neighbouring node is in a different quadrant at the same level;
- (iii) Neighbouring node is in a different quadrant at a higher level;
- (iv) Neighbouring node is in a different quadrant at a lower level.

Examples of each of these instances are illustrated below in Figure 7.12.

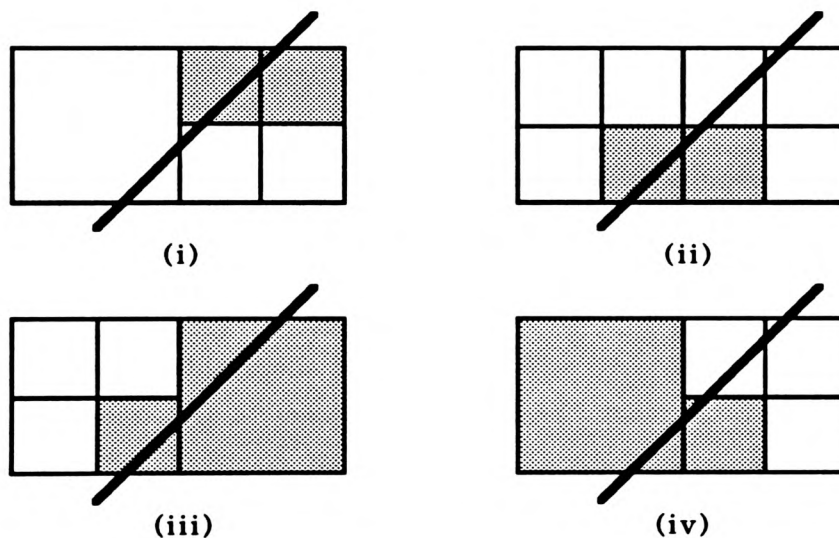


Figure 7.12 - Examples of the Four Possible Instances of Adjacency between Quadtree Nodes.

In general, the key of an adjacent node at the same quadtree level can be determined from the indexing scheme of the linear quadtree, or by using simple modular arithmetic. The quadtree needs to be searched to check whether this derived key actually exists at that level of decomposition. This will identify all of the adjacency instances of types (i) and (ii). However, if the derived key is not found, the adjacent node is at a different level of the quadtree. If it is at a higher level, the adjacent node will be identical to the derived key up to the position of

the most significant marker. If no key is found, the adjacent node is at a lower level of the quadtree. In this instance, the point of profile intersection will determine the key of the quadtree node at the next lowest level. This continues until the adjacent node at the lowest level is discovered.

Searching the quadtree is performed using a binary search. A sequential search is too inefficient, since at each node the compressed key may need to be decoded into its individual base five digits. However, the time required to perform each binary search is still significant. Decoding the keys is very time consuming, since an average of seven keys are decoded every millisecond. Hence the search for one adjacent node may take up to two milliseconds within the 5 metre surface patch quadtree for ST08. A typical profile though, may require access to hundreds of nodes. The total time required to interpolate a profile (which includes following the profile path; calculating intersections and directions of adjacent nodes; determining adjacent keys; searching for adjacent keys; and interpolating profile points) may require up to half a second of CPU time within a dense surface patch quadtree.

Direct profile interpolation within a compressed linear quadtree is too inefficient to be considered viable. Too much time is required to follow the path of the profile through the nodes of the quadtree. More specifically, the search for any given adjacent node is hindered by the need to decode a number of individual key elements. This is a direct consequence of using a compressed two-byte linear key. If the key was stored in an uncompressed format, the profile interpolation algorithm would be quite efficient, but storage costs would be excessive. In effect, the surface patch quadtree is typical of a DEM which can be either storage efficient or computationally efficient, but not both at the same time.

An alternative approach is to decode all the keys when the surface patch quadtree is first read. This takes approximately 1.2 seconds of CPU time for the 5 metre tolerance surface patch quadtree of data set ST08. However, the additional time needed to retrieve the coordinates of these surface patches and re-interpolate the 256x256 elevations on the original 50 metre grid DEM is only about 0.49 seconds for the densest quadtree. The time taken for each of these operations and the total time for quadtrees of up to 7800 nodes (36% of grid storage) is illustrated below in Figure 7.13. This graph clearly illustrates the high computational overheads of decoding the linear key. This accounts for 60% to 70% of the total quadtree to grid conversion time. One interesting feature of this graph is the time taken to re-interpolate the original 256x256 grid values. In all cases this is accomplished in under 0.2 seconds. This clearly illustrates the benefits of using a temporary data structure such as the regular grid, if it can be retrieved using a simple approximation method, such as bilinear interpolation.

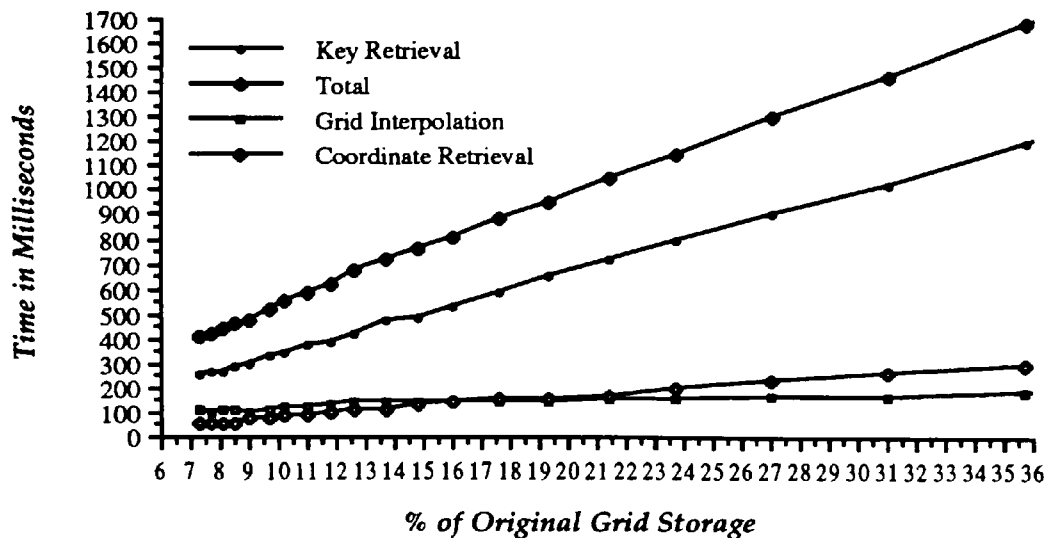


Figure 7.13 - CPU Time in Milliseconds for Quadtree Key Retrieval, Coordinate Retrieval, 256x256 Grid Interpolation and Total Quadtree to Grid Conversion Time.

Despite the high overheads of key decoding, conversion of the surface patch quadtree to a regular grid DEM is the most computationally efficient method for interpolating profiles from such a data structure. The 'one-off' overhead of grid retrieval is a maximum of 1.7 seconds (for the 5 metre quadtree of ST08). This is equivalent to the time taken for the interpolation of four average profiles directly from the compressed surface patch quadtree. Any alternative approach will always require the linear key to be decoded. Therefore, profile interpolation within the surface patch quadtree was accomplished with the use of the temporary regular grid data structure. The corresponding computational overhead of performing this task can be determined from Figure 7.13. However, average profile interpolation time is then the same as for the regular grid DEM (ie. less than 0.3 ms).

Once the profiles had been interpolated from the surface patch quadtree (via the temporary regular grid), the radio path loss prediction algorithm was applied. However, the random set of 1000 profiles are not the same as for the other DEMs due to the nature of the 256x256 subsets. The profiles are also shorter than the average of 10-12 kilometres for the full 401x401 data sets (ie. approximately 7 to 8 kilometres). Hence radio path loss estimation in the surface patch quadtrees has a number of factors which hinder a direct comparison with the results for other DEMs. Tables 7.5 & 7.6 overleaf, present the radio path loss results obtained for ST06 and ST08. These results have been averaged over the four 256x256 subsets covering the original data sets, thus representing 4000 different profiles.

Largest Abs. Error	Frequency = 200 MHz				Frequency = 400 MHz				Frequency = 600 MHz			
	A.A.v. Error	RMSE	No. of Profs ≤ 6dB 12dB		A.A.v. Error	RMSE	No. of Profs ≤ 6dB 12dB		A.A.v. Error	RMSE	No. of Profs ≤ 6dB 12dB	
25 m	3.700	5.218	799	965	3.896	5.906	764	936	3.881	5.917	752	934
20 m	3.091	4.472	854	978	3.124	4.878	821	964	3.088	4.844	819	965
15 m	2.498	3.702	901	988	2.359	3.822	877	984	2.283	3.729	882	987
10 m	1.815	2.923	945	992	1.617	2.824	932	996	1.536	2.702	939	997
5 m	1.056	1.964	977	997	0.878	1.812	973	999	0.831	1.734	976	1000

Table 7.5a - ST06 Radio Path Loss Errors for Surface Patch Quadtrees at 200, 400 & 600 MHz.

Largest Abs. Error	Frequency = 900 MHz				Frequency = 1400 MHz				Frequency = 1800 MHz			
	A.A.v. Error	RMSE	No. of Profs ≤ 6dB 12dB		A.A.v. Error	RMSE	No. of Profs ≤ 6dB 12dB		A.A.v. Error	RMSE	No. of Profs ≤ 6dB 12dB	
25 m	4.437	6.795	701	903	5.061	7.848	680	863	5.418	8.457	668	839
20 m	3.470	5.501	772	943	3.939	6.349	751	915	4.214	6.856	740	893
15 m	2.543	4.217	840	977	2.883	4.878	828	960	3.082	5.275	818	944
10 m	1.688	3.040	910	992	1.905	3.502	904	987	2.033	3.786	898	977
5 m	0.911	1.946	960	999	1.020	2.218	957	997	1.084	2.391	956	995

Table 7.5b - ST06 Radio Path Loss Errors for Surface Patch Quadtrees at 900, 1400 & 1800 MHz.

Largest Abs. Error	Frequency = 200 MHz				Frequency = 400 MHz				Frequency = 600 MHz			
	A.A.v. Error	RMSE	No. of Profs ≤ 6dB 12dB		A.A.v. Error	RMSE	No. of Profs ≤ 6dB 12dB		A.A.v. Error	RMSE	No. of Profs ≤ 6dB 12dB	
25 m	3.845	5.866	796	940	3.750	5.724	784	938	3.496	5.691	789	938
20 m	3.246	5.105	847	954	2.954	4.808	835	962	2.852	4.769	840	960
15 m	2.654	4.275	892	973	2.269	3.958	884	977	2.157	3.894	891	976
10 m	1.947	3.499	933	982	1.653	3.107	928	988	1.523	3.050	931	987
5 m	1.035	2.468	970	990	0.739	1.756	976	997	0.664	1.714	978	998

Table 7.6a - ST08 Radio Path Loss Errors for Surface Patch Quadtrees at 200, 400 & 600 MHz.

Largest Abs. Error	Frequency = 900 MHz				Frequency = 1400 MHz				Frequency = 1800 MHz			
	A.A.v. Error	RMSE	No. of Profs ≤ 6dB 12dB		A.A.v. Error	RMSE	No. of Profs ≤ 6dB 12dB		A.A.v. Error	RMSE	No. of Profs ≤ 6dB 12dB	
25 m	3.640	6.014	783	925	4.001	6.729	767	903	4.236	7.177	758	888
20 m	2.960	5.082	828	950	3.246	5.689	814	930	3.436	6.070	807	919
15 m	2.205	4.094	886	970	2.399	4.567	876	959	2.537	4.869	869	952
10 m	1.540	3.190	929	984	1.674	3.539	922	976	1.769	3.776	918	972
5 m	0.707	1.914	972	995	0.781	2.164	969	992	0.826	2.316	968	989

Table 7.6b - ST08 Radio Path Loss Errors for Surface Patch Quadtrees at 900, 1400 & 1800 MHz.

A significant feature of Tables 7.5 and 7.6 is the variation in radio path loss performance between different frequencies, particularly for ST06. For example, in the ± 25 metre quadtree, there is a significant difference between results at 200 MHz and 1800 MHz (R.M.S. errors of 5.218 and 8.457 dBs, respectively). However, all such occurrences tend to occur when the storage costs of the surface patch quadtree are very low, such as less than 10% of the storage of the original regular grid. More consistent results are obtained for the more error-constrained quadrees. At these higher storage levels, the radio path loss performance at the various examined frequencies is also consistent with that of the other DEMs (ie. the best overall radio path loss estimation is generally at 600 MHz, followed by 400 MHz, 900MHz or 200 MHz, 1400 MHz and 1800 MHz).

The graphs of average radio path loss R.M.S. errors for ST06 and ST08 are presented below in Figure 7.14 and the average number of profiles within an error of ± 6 dBs and ± 12 dBs in Figure 7.15. It should be remembered that the use of such surface patch quadtrees is not recommended for models which reduce the storage costs of the regular grid DEM by more than 90%. For this reason, the results presented in Figures 7.14 and 7.15 are too unstable for less than 10% of grid storage, particularly at higher frequencies (for example, 1800 MHz).

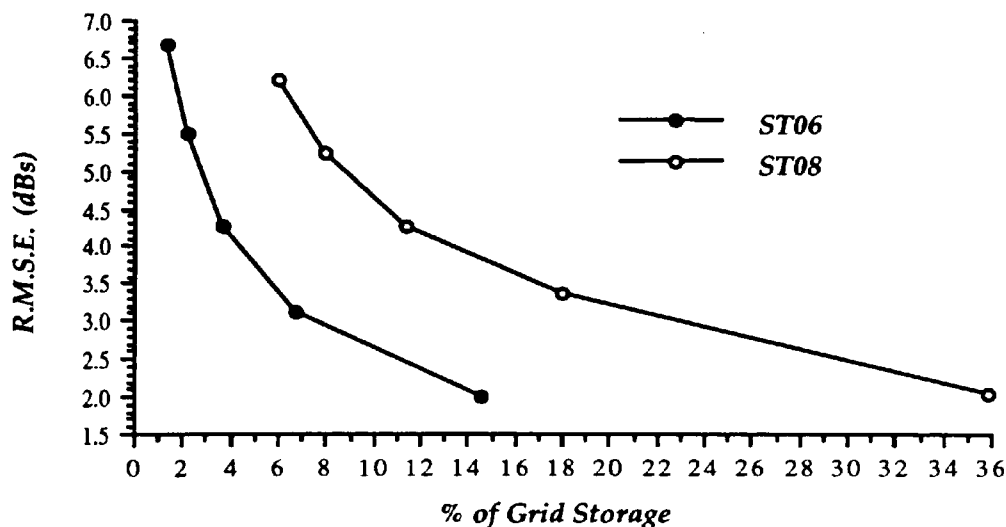


Figure 7.14 - R.M.S. Errors (dBs) of Radio Path Losses for Surface Patch Quadrees Averaged Over the Frequencies of 200, 400, 600, 900, 1400 & 1800 MHz.

The graph of radio path loss R.M.S. errors (Figure 7.14) behaves in a similar manner to the graph of elevation errors. In Section 7.4, it was concluded that the overall surface patch quadtree elevation errors were very similar to those of the polynomial models. This relationship is surprisingly also reflected for radio path loss performance. The largest elevation errors for the polynomial DEMs occur in areas of steep slope and in a displacement of critical features, such as ridges and peaks. This was due to the limited flexibility of the

surface functions in relatively large surface patches. However, whilst the largest elevation errors for the surface patch quadtree are more constrained (due to its ability to adapt to the terrain variability down to the original grid sampling density), there are generally many more small errors. A quadtree node is represented by a bilinear surface joining its four corner elevations, whilst the same patch may be more efficiently modelled with a least-squares polynomial. Therefore, in terms of estimating radio path loss, the polynomial DEMs will perform better than the surface patch quadtree in slightly variable terrain, whilst the surface patch quadtree is better for rougher terrain.

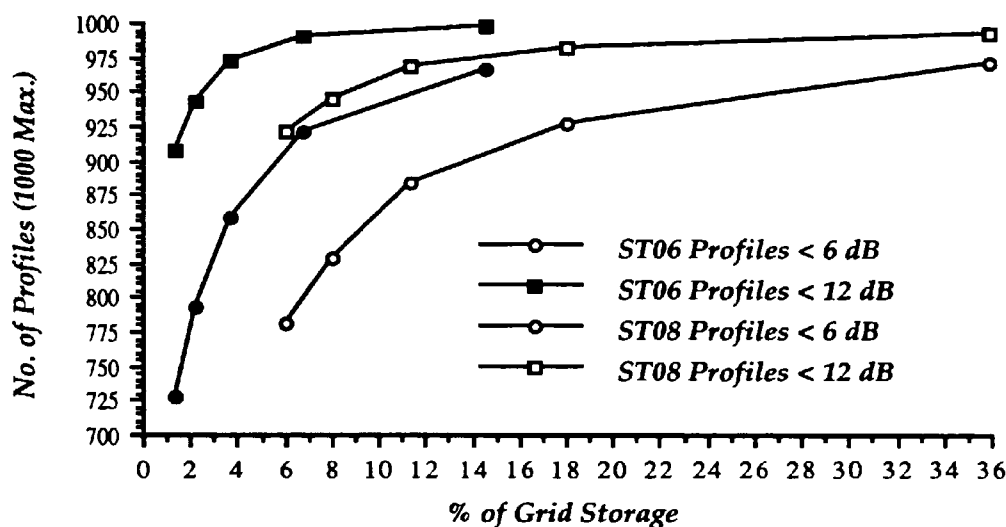


Figure 7.15 - No. of Surface Patch Quadtree Profiles Within Path Loss Error of 6 & 12 dBs Averaged Over the Frequencies of 200, 400, 600, 900, 1400 & 1800 MHz.

7.6 Chapter Conclusions

The greatest advantage of the surface patch quadtree is its flexibility in adapting to terrain variability, and in particular rough terrain. It is relatively simple to construct and yet it encompasses many of the advantages of a grid-based DEM (ie. variable density grid), whilst overcoming the data redundancy problem associated with the regular grid. The use of a linear quadtree allows for a storage-efficient indexing of the surface patches in a more flexible manner than for the variable density grid DEM discussed in Chapter Four. The extensive storage savings that are obtained are accomplished with the use of very simple planar, linear or bilinear functions which can represent a surface at grid intervals of between 50 metres and 6.4 kilometres. The use of more flexible surface functions does not meet the storage-efficient requirement of being able to model the terrain at scales up to the original sampling density.

The surface patch quadtree was examined in order to identify a more efficient DEM than the

polynomial surface patch DEMs discussed in Chapter Six. Since the quadtree adopts a different approach to surface modelling than the adaptive polynomial DEM (ie. variable patches of uniform surface functions as opposed to uniform patches of variable functions), it was initially thought that the cause of the largest radio path loss errors in the latter model could be overcome. However, whilst the surface patch quadtree is more constrained to the original gridded surface in critical regions (for example, valleys, ridges and peaks), the adaptive polynomials are better suited for less variable terrain. As a result, the overall performance of both types of DEMs are very similar. 95% of terrain profiles can be obtained within a radio path loss error of ± 6 dBs (for ST08) at a cost of just over 30% of the original 50 metre grid DEM storage. Whilst this may seem satisfactory, the additional computational overheads in retrieving a temporary regular grid for profile interpolation would suggest that the surface patch quadtree is not a viable alternative to the 50 metre regular grid DEM.

An alternative surface patch quadtree data structure for storing regular grid DEMs is the elevation pyramid, which can hierarchically encode continuous surfaces with minimum overhead. In particular, two such data structures have been described by Dutton (1983) and Shaffer (1989). Dutton's DEPTH model (Difference-Encoded Polynomial Terrain Hierarchy) compresses regular grid DEMs to an equivalent of 2.666 bits per elevation (83.33% storage saving over the 50 metre grid), using a tolerance-sensitive relative coding scheme. Each cell of the pyramid or linear quadtree contains a code which represents the difference between the cell's elevation and the elevation above it in the pyramid. Hence the vertical domain is approximated (polynomially) at the same time as the horizontal domain. Shaffer's elevation pyramid is constructed in a similar manner, such that the elevation at any pixel is calculated by traversing a path from the root to the pixel by refining a local elevation at each stage by interpreting two bit codes stored with each node. This representation requires an average of less than 3 bits per elevation (81.25% storage saving). Shaffer states that 'this is a lossless encoding', but with the stipulation 'when the difference between sibling pixels is not "too great"'. However, it is the definition of "too great" which affects the practicality of such a representation. Dutton's model is tolerance-constrained, but its efficiency is also limited by terrain variability.

Prototype terrain models were constructed for both Dutton's and Shaffer's pyramids, but both proved to be impractical as an efficient representation for ST08 and to a lesser extent ST06. Whilst elevation pyramids may remove data redundancy, the degree to which they can adapt to terrain variability is severely limited. As such, they offer little or no improvement over the Differential Altitude Grids and Huffman-Encoded Grids discussed in Chapter Five.

Chapter Eight

Irregular Point DEMs

8.1 Introduction

In order to represent a topographical surface without information redundancy, the data structure should support an arbitrary arrangement of points which represent the significant features of the terrain, possibly as an irregular grid. Such models provide the best approximation of the topographic surface since they describe more closely the framework of the terrain and best express its geomorphology (Neumyvakin & Yakovlev, 1986).

However, the availability of such data is limited for most users, since the majority of DEMs are in the form of a regular grid. With the widespread availability of this data, there has been extensive research undertaken to derive surface-specific features from dense grid DEMs, in order to formulate irregularly sampled terrain models, or to simulate drainage basins/river networks (Peucker & Douglas, 1975; Fowler & Little, 1979; Mark, 1983; Jenson, 1985; Burrough, 1986; Douglas, 1986; Chen & Guevara, 1987; Jenson & Domingue, 1988; Lee, 1989; Skidmore, 1990). The storage savings that can be attained using this form of surface representation are quite significant. Provided the original grid is of a sufficient density and accuracy, it is reasonable to suggest that critical features can be identified from it. These points may then be used as the raw data for variable resolution or irregular point DEMs (Devereux, 1985; De Floriani et al., 1985). The most common irregular DEM is the triangulated irregular network or TIN (Peucker et al., 1978), which represents the surface as triangular facets of irregular size and shape, thus maintaining a topological relationship between points.

The critical points within an irregular DEM (eg. vertices of a TIN) represent the geomorphic features that characterise terrain, such as channels, ridges, peaks, pits, passes and saddles. However, this topography can be locally extracted from regular grids of fine resolution at a cost of incorporating only the small inherent errors in the spatial coordinates. The magnitude of such errors depends upon the sampling density or resolution of the original grid. For example, a peak near the centre of a cell is represented by one of the four grid vertices at a lower elevation and displaced by a distance of up to $1/\sqrt{2}$ of the sampling interval. By virtue of the extensive use of regular grids as terrain models, such errors are considered tolerable for most applications. By the same argument, the accuracy of grid data can be considered sufficient for the construction of an irregular point DEM, such as a TIN (eg. Fowler & Little, 1979; De Floriani et al, 1983; Chen & Guevara, 1987; Douglas, 1986; Lee, 1991). This approach allows the extensive grid DEMs that are available to be used as a source for other models without the prohibitive data acquisition costs.

The main elements of irregular point DEMs are the x , y and z coordinates of the data points, with their topological relations identified implicitly, explicitly or algorithmically (Peucker, 1978). Implicit relations are defined by the internal order of the elements; explicit relations are stored as part of the records; and algorithmic relations are computed as part of the data manipulation. Peucker distinguishes these algorithms as those which are independent of topology, local algorithms and tracking algorithms. However, as an irregular distribution of points does not have implicit topological relations, the data may either be transformed into a data structure which has implicit relations, such as a regular grid, or the relations may be added explicitly, or alternatively they could be determined when required. As will be shown for the triangulated irregular network in this chapter, the degree to which explicit topological relations are maintained within the data structure is very significant in terms of storage and computing efficiency.

This chapter examines some of the techniques for extracting information rich data from regular grids, for the specific use of forming irregular point DEMs. In particular, the benefits of using the TIN have been analysed with respect to the regular grid DEM, together with the degree to which topological relationships should be contained within the data structure itself. An alternative storage-efficient TIN is considered, termed the implicit TIN, for which no topological relations are stored. However, the adjacency relationships are reconstructed in local areas of interest by application of a Delaunay triangulation algorithm to the points or TIN vertices. This data structure is also considered for local interpolation techniques, without the need for maintaining or retrieving any topological relationships between points.

8.2 Grid Information Rich Points

Points, lines and regions are termed 'high-information' when they represent the characteristics of terrain most effectively, and are considered the minimum set for reconstruction of the original surface. A prerequisite to identifying terrain characteristics requires a mathematically sound, unrestricted definition of these features (Douglas, 1986). Peaks and pits are surface maxima and minima; passes are maxima in one direction and minima in the other; ridges are maxima connecting peaks with passes; and channels are minima connecting pits with passes (Peucker & Douglas, 1975). Douglas (1986), gives an extensive overview of these definitions from the relevant literature, and states that:

"Peaks, pits and passes represent points on a surface that contain by virtue of their definition much more information about the surface than other points. Beyond the mere location of itself as a point in three-dimensional space, a point tagged as a peak contains information that all points in the immediate location of itself are lower than itself. Similarly, a pit contains information that all points in the immediate neighbourhood are higher. Ridge and channel lines are similarly information rich. Ridge and channel lines are slope lines that are in regions of higher convexity or concavity, respectively. They are local phenomena that may be identified locally."

An effective method for identifying ridges and channels is to pass a 3x3 kernel or pixel window over the grid, centred on the node to be tested, searching for local maxima and minima in the four orthogonal and diagonal directions (Jenson, 1985; see Figure 8.1a). A simpler approach is to flag the lowest or highest node of a grid cell (ie. 2x2 pixels), such that all unflagged nodes will represent ridges or channels (Douglas, 1986). Both methods are equally efficient, requiring just a single pass through the grid model. However, these algorithms tend to produce clouds of pixels, rather than unambiguous single pixel wide lines. The generally accepted approach to reducing the selected points is to use a line thinning algorithm to produce skeletons of the ridges and channels (Peuquet, 1981; Pavlidas, 1982; Jenson, 1985; Douglas, 1986; Greenlee, 1987). Individual line segments may then be identified, such that the topology of the adjacency relationships of all lines can be determined. Two other operations ancillary to these processes are line smoothing and spike and gap removal (Peuquet, 1981).

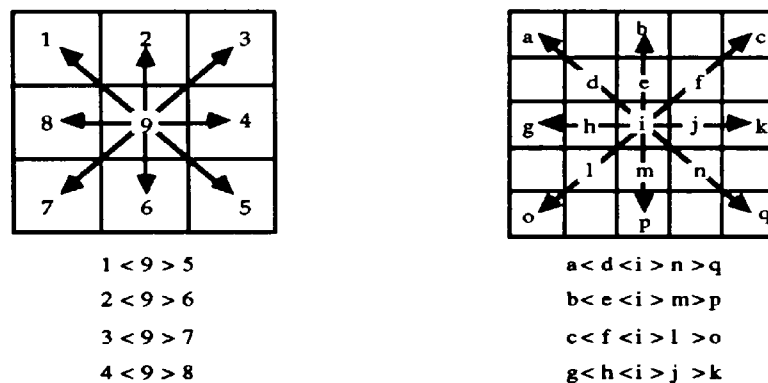


Figure 8.1a & 8.1b - Jenson's 3x3 Ridge Kernel and Extension to 5x5 Kernel

(eg. In 8.1a, the elevation at 9 is a local maximum if it is greater than the heights at 1 & 5).

Peucker and Douglas (1975) use a local 3x3 operator to classify every point of a surface by an analysis of its neighbours. However, they demonstrate that feature recognition is dependent upon the tolerances and selection criteria of the user, and any algorithm can be highly sensitive to noise, such that preprocessing or smoothing of the data is essential. The ideal method is one in which the potential points and lines are detected, and then improbable points are eliminated and missing points added.

Chen and Guevara (1987) discuss a different approach to point selection, in which a systematic selection of very important points (VIP) is used to construct a TIN. A significance is assigned to each grid node, indicating the importance of contribution to the representation of the surface. A node is selected, only if it cannot be predicted from the values of its neighbours. The measure of change from its neighbours can be made with a high-pass filter, such as a spatial differential or Laplacian operator. As such, the percentage of points to be selected from the grid DEM can be pre-specified by the user, or alternatively points can be selected if their significance is above or

below an upper or lower tolerance threshold.

Makarovic (1983) considers a similar approach to Chen & Guevara using a number of variations of the Laplacian operator, but for hierarchically structured grid data. This is of greater benefit for very dense grid data, such as photogrammetrically acquired data. The effectiveness of the method for compressing vertices depends mainly on the terrain roughness and the magnitude of tolerance threshold used.

However, with most applications for detecting critical points, the performance may be affected by a significant degree of 'noise' within the data. In some instances, this effect can be reduced by extending the size of the kernel, such as a 5x5 pixel window (Figure 8.1b) or defining tolerances within the selection criteria. An alternative approach for reducing the effect of noise is to smooth or generalise the grid DEM before delineation (Davis et al, 1982; Loon, 1984). Smoothing may be defined as a spatial process designed to modify the significance of certain types of local terrain features whilst maintaining the regional shape (Davis et al, 1982). The most popular approaches to smoothing are iterative filtering with a spatial convolution operator (eg. Laplacian or bi-harmonic) and weighted least squares filtering. The latter approach is perhaps the simplest, since it involves a linear transformation using a filter consisting of a series of weights, usually based on a 3x3 kernel. This weighted moving average should preserve the mean value of the original grid DEM, thus being 'phase-distortionless'. Loon (1984) states that to achieve this, the weights of the filter must be isotropic or symmetrical about the principal weight. One such function is the Gaussian function:

$$c(d) = e^{-a^2 d^2}$$

where a is some constant and d is the squared distance in grid units from the central point.

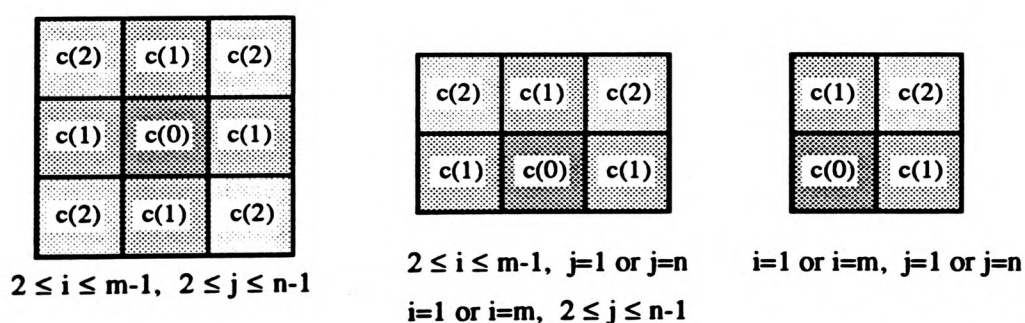


Figure 8.2 - Example of a 2-D Isotropic Weighted Filter for All Cases Within an $m \times n$ Grid.

Example values may be $a = 0.0$ (all weights equally assigned); $a > 4.0$ (principal weight = 1.0); or $a = 0.83$, which approximates the binomial weighted smoothing filter (Figure 8.3).

.0628	.125	.0628
.125	.2489	.125
.0628	.125	.0628

.0837	.1668	.0837
.1668	.3322	.1668

.2225	.1117
.4432	.2225

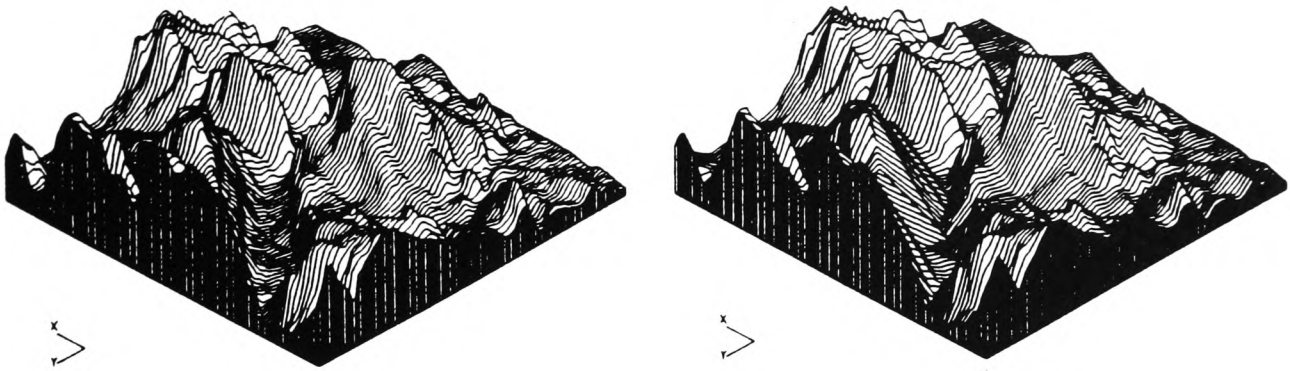
Figure 8.3 - Gaussian Filter (with $a=0.83$).

This function can be used with suitable weights to smooth the regular grid DEMs before application of the delineation algorithms. The results produced by these algorithms are presented in greater detail in *Appendix C - Surface Specific Points*. These results show that the percentage of selected critical points is somewhat arbitrary, and varies considerably between methods. However, an advantage of Chen & Guevara's algorithm (1987), is that the user may specify this number of points. An important consideration which therefore arises within the point selection task is the determination of 'what is the optimum number of critical points to be selected from a DEM, which produces an efficient alternative surface model or TIN'. It is apparent that the solution is subject to the nature and variability of each individual terrain surface. As such, it is impossible to determine this optimum value without detailed analysis of the terrain beforehand.

Weibel (1989) reports on the design and implementation of a strategy for terrain generalisation which is adaptive to different terrain types, range of scales and various map purposes. Such an implementation allows for greater flexibility and better evaluation. Future developments in extracting critical points from gridded DEMs will require a greater understanding or interpretation of topography. Weibel & DeLotto (1989) examine some of the difficulties of performing a classification of topography.

Even if the optimum number of selected points is known, it cannot be guaranteed that any algorithm will determine these points correctly or most efficiently. Most delineation algorithms require filtering and generalisation to further remove redundant points. The degree to which this is accomplished can be predetermined by the user to meet the requirements of the number of selected points. Only when this optimum number of points have been selected and the model generated, may the overall performance of the critical points within the surface model be determined. However, results have shown that some large errors may occur within TINs and other irregular DEMs, even if a large proportion of grid points are selected. For example, valley floors may be generalised to a thickness of one pixel, resulting in large errors occurring along the valley sides (see Figure 8.4, below). One feature common to most point selection algorithms is the lack of control over elevation error in the irregular surface model. The overall performance of the model cannot be predetermined, such that the relative contribution

of each point is unknown with respect to the maximum or average error.



*Figure 8.4 - Isometric Projection of a Grid DEM & Corresponding TIN of Selected Critical Points
(The variation in error is most prominent along the valley floor in the foreground).*

Hence, there is no ideal method (requiring one or two passes of the data) which detects critical points or lines, eliminates superfluous data and supplements critical points that are omitted to within a maximum tolerance. In practice, an iterative procedure can be applied, whereby points are successively selected until a pre-defined error tolerance is satisfied. Such approaches, termed hierarchical triangulation, define an initial set of 'very' critical points, (using some of the algorithms described above or in the simplest case, the four corner points), which when triangulated are supplemented by the insertion of the unused point of largest error (Fowler & Little, 1979; De Floriani et al, 1983).

Scarlato & Pavlidis (1991) propose an adaptive hierarchical triangulation which is error-constrained and related to the surface characteristics derived from a regular grid. The errors and their positions determine if and how the triangles will be split. Kumler (1990) reports that TINs whose vertices are selected from digitised contours and spot heights produce smaller errors than grid-derived TINs, even though both structures were derived from the same original data and require the same amount of storage space. However, the more widespread availability of regular grid data sets and the ease to which accuracy of a derived TIN can be determined from it suggests that grid-derived TINs will remain popular.

The construction of the TIN used in this study, consists of a sparse set of critical points derived from a grid DEM using the algorithms of Peucker & Douglas (1975) and Chen & Guevara (1987). The choice of initial critical points is arbitrary to some extent, but these algorithms allow thresholding tolerances to adjust this selection process to individual preferences. A Delaunay triangulation of these initial points is then constructed using the criteria set out in the algorithm of McCullagh & Ross (1980). The original grid DEM is interpolated from this TIN

such that the errors associated with all unused points can be determined. The point with the largest error or priority is then inserted into the TIN using a local re-triangulation and the list of priorities is updated accordingly. De Floriani (1987), Correc & Chapuis (1987), Daly (1989), Heller (1990) and Kao et al (1991) describe algorithms to accomplish this, together with further consideration of some of the implementation issues. De Floriani & Puppo (1988) consider line insertion as well as point insertion in their constrained Delaunay triangulation (CDT).

Point insertion continues until the TIN satisfies the error constraints imposed upon it by the user. This approach has a tendency to predominantly select critical ridge and channel points as TIN vertices, at a cost of excluding local maxima and minima (ie. peaks and pits) and less significant ridge and channel points. This may hinder the efficiency of the TIN slightly, for visualisation applications such as profiling, line-of-sight calculations and radio path loss calculations, but the maximum absolute elevation error will always be constrained. However, the effects of missing critical points (from the point insertion algorithm) can be minimised to a great extent by increasing the number of initially selected critical points.

The data structure used for the creation of the TIN must support the flexibility required for dynamic triangulation. An advantage of dynamically selecting TIN vertices is that the surface can be represented at different levels of resolution or error-constraints in a similar manner to the Delaunay pyramid of De Floriani (1989). For example, an initial TIN at the top level may be supplemented by sets of points at lower levels which constrain the triangulation to maximum errors at say, 10 or 5 metre intervals.

This approach of hierarchical triangulation has recently received some criticism (Lee, 1991), due to the fact that an inserted point might become redundant if another point is inserted 'nearby' at a later stage. The reasoning behind this is that the surface could be represented to the required tolerance by one or other of the points. Hence, there may be some data redundancy in this form of point selection. This is particularly true when the set of initial critical points (before triangulation and point insertion) is large.

Lee (1989, 1991) suggests an alternative approach termed the 'drop heuristic method', in which the initial triangulation consists of every original grid point within the DEM. Points which cause the least elevation difference in the TIN are dropped in an iterative procedure which also refines the triangulation at each stage. Lee (1991) compares this approach to hierarchical triangulation (De Floriani et al, 1985) and the filter method of Chen & Guevara (1987). Lee's results have shown that the heuristic approach produces smaller elevation differences (in terms of totals, means and standard deviations) for all six of his tested DEMs and for equivalent numbers of TIN vertices (ie. 10% of grid points). However, no results are given for

smaller amounts of TIN vertices and no evaluation is made as to how the efficiency of the heuristic algorithm will depreciate with respect to the other algorithms for fewer selected points.

The heuristic approach will initially require a significant amount of RAM memory, if the TIN is to be formulated for every regular grid vertex. The computational efficiency of such an approach may also make the algorithm unattractive for large DEMs. An additional disadvantage, which is inherent in all TINs derived from regular grid DEMs, but which is more apparent in the heuristic approach, is the degeneracy problem, especially with algorithms such as a Delaunay triangulation algorithm. Degeneracies occur when more than three points in the plane are co-circular, such as for example, the four vertices forming a square or rectangular grid cell. The choice of diagonal which forms the two triangles may be critical, yet an arbitrary decision may create large interpolation errors. This problem is addressed in greater detail in Section 8.3.

Results from all grid extraction algorithms, including the recent work of Lee (1991), suggest that there is plenty of scope for future work in the derivation of TIN vertices from regular grid DEMs. This is particularly apparent for the determination of the 'optimal' set of points which will represent a TIN surface to a required accuracy standard (eg. maximum, average or root mean square error). Each method has its own advantages and disadvantages and each has a tendency to select points of certain types of topographic features. Lee (1991) examines this in greater detail for the heuristic, hierarchical and filter methods and concludes that users should be aware of this fact and the strengths and weaknesses of each algorithm for any specific application.

8.3 The Delaunay Triangulation

Methods of constructing the planar triangulation have received widespread attention in the field of surface modelling (Gold, 1979). Heller (1986) presents an overview of triangulation algorithms and gives an extensive bibliography with abstracts. For most applications, such as interpolation, a good triangulation produces triangles which are as equiangular as possible, thus avoiding elongated triangles. McCullagh (1987) states that a triangulation should have the properties of stability, equilateralness and non-intersection for some applications, such as contouring, where an arbitrary triangulation may not be acceptable. The Delaunay triangulation has become accepted as one of the best approaches for the creation of a TIN, since it satisfies these requirements and produces a unique solution in one pass of the data. De Floriani (1987) reviews the literature with respect to both static and dynamic algorithms for constructing the Delaunay triangulation. This triangulation is the dual of the net of Thiessen

polygons (also known as the Voronoi diagram, Dirichlet tessellation, proximal tessellation, "S" polygons or Wigner-Seitz cells), in which any location on the plane is assigned to the polygon containing the nearest data point (Gold, 1979). These polygons may be thought of as the cells of a growth process. Green & Sibson (1978) state:

"Tiles (polygons) which have a boundary segment in common are said to be contiguous, as are their generating points. In general tiles meet in threes at vertices, so the lines joining contiguous generating points define triangles; these triangles can easily be shown to fit together into a triangulation of the convex hull of the generating points; the perpendicular bisectors of the edges of this triangulation give the boundaries of the tiles, and the circumcentres of the triangles are the vertices of the tiles. The triangulation is called the Delaunay triangulation."

The properties of such tessellations have been extensively used in procedures for the statistical analysis of spatial patterns. The diversity of names by which the Delaunay tessellation and its dual are known is primarily due to its widespread use in a number of different fields. These include astronomy, biomathematics, computer science, geography, meteorology, metallurgy, numerical analysis and packing and covering. Lee & Schachter (1980) give a number of references for each of these applications.

Three lemmas can be distinguished which globally and locally define a Delaunay triangulation (Lee & Schachter, 1980):

Lemma 1: For any triangulation of N nodes, B of which are on the boundary (convex hull), there are $2N-B-2$ triangles and a total of $3N-B-3$ edges (Euler's Theorem).

Lemma 2: Two vertices form a Delaunay edge, if and only if there exists a circle passing through the vertices that does not contain any other vertex.

Lemma 3: Three vertices form a Delaunay triangle if and only if its circumcircle does not contain any other point in its interior.

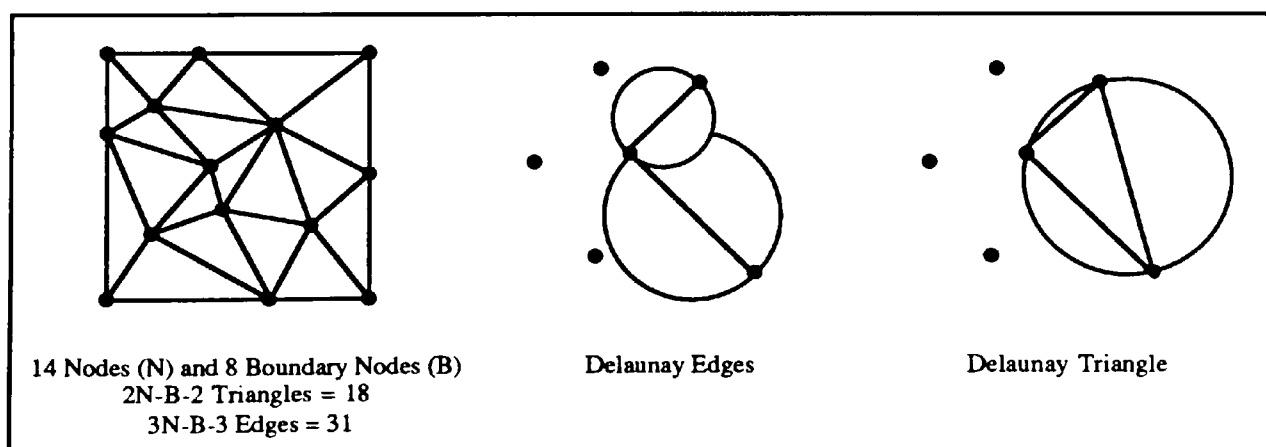


Figure 8.5 - Illustration of the Three Lemmas Defining a Delaunay Triangulation.

From Lemma 2 and 3, a simple algorithm can be defined for the construction of the Delaunay triangulation, in which the properties of Lemma 1 are implicitly incorporated. For small data

sets, Verts (1991) describes an 'extremely simple Delaunay triangulation algorithm that runs in $O(N^4)$ time, which is slow and dirty'. However, the algorithm used in this study is based upon that of McCullagh & Ross (1980). The search of the neighbours of a vertex (Thiessen neighbours) proceeds in a clockwise direction around that point. For any known neighbour, the next neighbour is located as the vertex for which a circumcircle passes through the three points, with no other point inside the circle (Lemma 3). This is accomplished with a search circle passing through the vertices of the known edge, checking for an inscribed point in a clockwise direction. If no points are found, the size of the circle is increased, whilst if more than one point is located, the Thiessen neighbour which has the largest angle subtended from the known edge is selected. The process of calculating the Thiessen neighbours or Delaunay edges of a point is illustrated overleaf in Figure 8.7. This assumes that the first edge is known and the search for each Thiessen neighbour uses the known edge as the basis for forming the search circle (ie. Premise: Let the diameter of the initial search circle be equal to the known edge joining the rotation point [R] and the known neighbour [N]). For any triangulation, the initial process is to calculate the convex hull of the points, such that a set of initial known neighbours can be established.

This algorithm is satisfactory for 'arbitrary' located data, but since more and more TIN data are derived from regular grid DTMs, the degree of arbitrariness may be insufficient for a consistent triangulation. The regular coordinates of the vertices may cause 'degeneracies' to occur, such that four or more points may lie on a circumscribing circle (ie. co-circular), each subtending the same angle from the known edge (Figure 8.6).

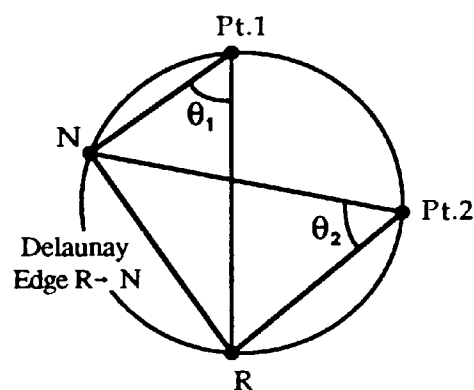
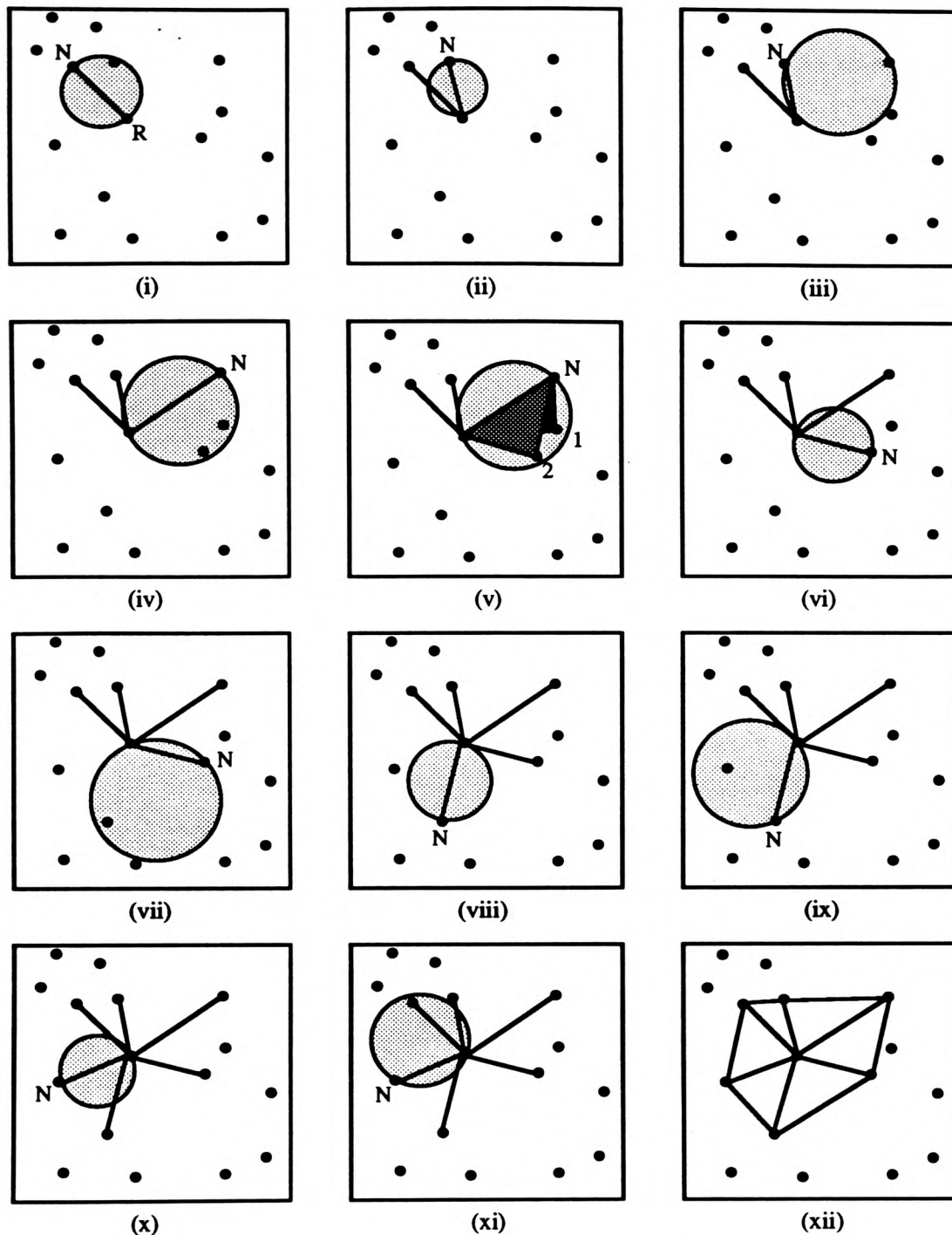


Figure 8.6 - Four Co-circular Points Creating a Degeneracy ($\theta_1 = \theta_2$).

Thiessen polygons will usually meet in threes, but Green & Sibson (1978), Sibson (1978), Bowyer (1981) and Watson (1981) have all recognised the fact that if four or more tiles meet at a vertex, then such a vertex is said to be degenerate. A regular grid determines a tessellation in which every vertex is said to be degenerate. In this instance, the dual triangulation could be formed from any of the two possible diagonals of each grid cell. Since there is a strong likelihood a TIN may be derived from a regular grid, it is possible that degeneracies may be quite common,



- | | | |
|--------|---|--|
| (i) | For the stated premise - One point found | \therefore New Known Neighbour |
| (ii) | For the stated premise - No points found | \therefore Increase Search Circle |
| (iii) | For increased Search Circle - One point found | \therefore New Known Neighbour |
| (iv) | For the stated premise - Two points found | \therefore Calculate Subtended Angles |
| (v) | Let N be the point with the greatest angle | \therefore New Known Neighbour = Pt. 2 |
| (vi) | For the stated premise - No points found | \therefore Increase Search Circle |
| (vii) | For increased Search Circle - One point found | \therefore New Known Neighbour |
| (viii) | For the stated premise - No points found | \therefore Increase Search Circle |
| (ix) | For increased Search Circle - One point found | \therefore New Known Neighbour |
| (x) | For the stated premise - No points found | \therefore Increase Search Circle |
| (xi) | For increased Search Circle - One point found | \therefore New Known Neighbour |
| (xii) | Since New Known Neighbour = Original Known Neighbour \Rightarrow Finish | |

Figure 8.7 - Illustration of the Process for Finding the Thiessen Neighbours of a Point (R).

especially if a large number of points are selected. This is a problem which seems to have been ignored to some extent in the literature, since for most applications the triangulation or tessellation algorithm is unlikely to be applied to such data (Green & Sibson, 1978).

By the above definitions, both diagonals of a degenerate quadrilateral such as a square or rectangle are valid edges, thus causing an overlapping or intersection of triangles (see Figure 8.8). If the edges are calculated from every vertex, the algorithm will produce an inconsistent triangulation.

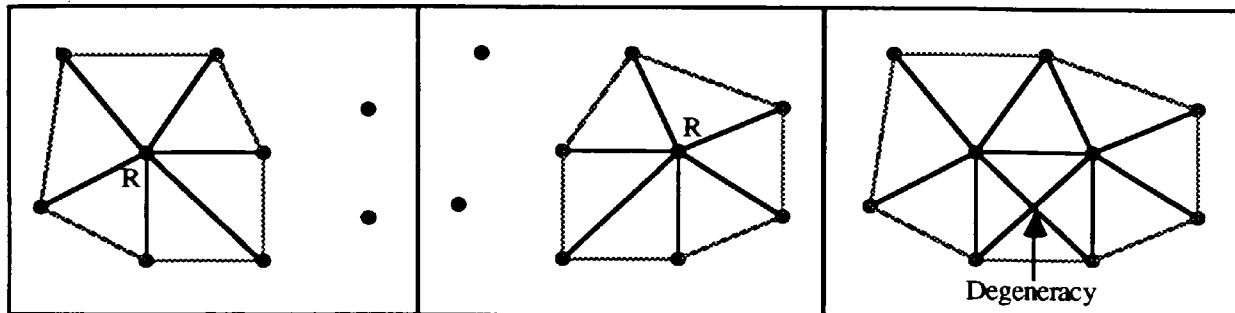


Figure 8.8 - Degenerate Triangulation for Two Rotation Points (R) .

Green & Sibson (1978) state that 'near-degeneracies' may cause an inconsistent tessellation (triangulation), but can be avoided by efficient programming. They believe 'that it is of little importance if for numerical reasons the diagonal contiguity is recorded incorrectly when the situation is close to degenerate', provided it is not stored inconsistently. However, it will be shown below that 'true-degeneracies' do occur in some applications and the selection of the correct diagonal can be critical. McCullagh & Ross (1980) suggest selecting the point closest to the known neighbour providing it is not also closest to the other vertex of the known edge. However, this is not sufficient for a degeneracy of the above type (Figure 8.8). Instead a local decision rule is required which takes into account the nature of the surrounding points, since an arbitrary choice of edge may cause large interpolation errors (see Figure 8.9). For example, the diagonals of the four points forming a grid square could represent either a ridge or a channel, but the true feature can only be determined from examining the surrounding, local terrain.

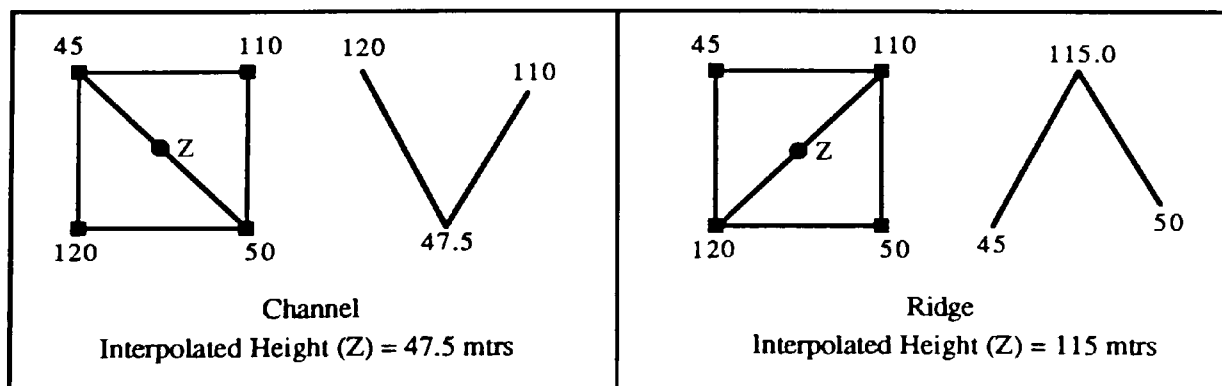


Figure 8.9 - 'Exaggerated' Example of the Significance of Selecting the Correct Edge for a Degenerate Triangulation.

Whilst this is an exaggerated example, tests carried out on the regular grid DEM ST08 have shown that for both choices of triangle edges, discrepancies of up to ± 25 metres were encountered for the interpolated centre point of each grid cell. Whilst the likelihood of such errors occurring for any derived TIN will be small for most applications, the user should bear this problem in mind. For such degeneracies, an examination of the local terrain to determine the true TIN edge can be computationally expensive. One method might be to fit a polynomial surface through the nearest n points, such that the true edge is the one that best fits the interpolated values from the polynomial. However, this form of three-dimensional TIN construction may be impractical for a TIN which has many degeneracies. Whatever solution is adopted, the triangulation must be consistent throughout.

This degeneracy problem highlights a major deficiency of Delaunay triangulation. Whilst the method is ideally suited to the triangulation of irregular points on the two-dimensional plane, extensions to more regular data in three-dimensions can create degeneracies or large errors. Furthermore, Delaunay triangulation algorithms may produce triangle edges which contradict the topology of the actual surface (Christensen, 1987; Scarlatos, 1989). Special consideration should be given for contours, cliffs, faults and other surface discontinuities and breaklines. De Floriani & Puppo (1988) consider the insertion of such lines into a constrained Delaunay triangulation. Kettelman (1987) also describes a post-formation technique in which all triangles are checked for breaklines, such that the common edges of the quadrilaterals forming the triangles may be swapped.

Spatial addressing of points is accomplished with a simple grid overlay or box-sort structure (McCullagh & Ross, 1980), which allows fast access to the TIN vertices. It assumes that the points have been pre-sorted in both the x and y directions. However, if the vertices are derived from a regular grid DEM, the data can be written directly to a file in this 'sorted' order. Each box or grid cell maintains a pointer indicating the position of the first point within the list of vertices, such that any number of points may reside in a box. Therefore, any point of known coordinates can be directly mapped into a box address, such that the vertices within that cell can be directly accessed from the vertex file. Search operations are augmented by a 'radius of search' parameter, which determines the range of neighbouring cells to be accessed. The size of the grid overlay will determine the optimality of searching operations. McCullagh & Ross (1980) suggest a grid which allows an average of four points per cell. An even denser grid may reduce search time further, but at a cost of increased storage for the pointers. It was found that a good compromise between search time and storage was three to five points per grid cell. Hodgson (1989) describes a similar method to that of McCullagh & Ross, termed the 'sortedcell matrix', which is used for rapid grid interpolation.

The properties of a grid cell addressing scheme can be utilised to compress the x , y and z

coordinates into two bytes of storage. This assumes that the vertices are derived from a regular grid DEM. Since the grid provides a direct mapping of coordinates into cell addresses and vice versa, the coordinates of vertices within these cells can be represented at a local level. Such an addressing scheme would relate the local coordinates to a fixed point within the grid cell, such as a corner. The feasibility of this approach is dependent upon the resolution of the grid overlay and the average number of points per cell. For the prototype implementation discussed, the local coordinates of each x and y value may be represented by three or four bits, with the elevation represented in eight to ten bits. In the example below (Figure 8.10), the grid overlay corresponds to 8x8 pixels, such that each x and y coordinate can be represented in three bits of storage. These values can be concatenated with each 10-bit elevation into a two-byte value, thus presenting a compact form of external file storage.

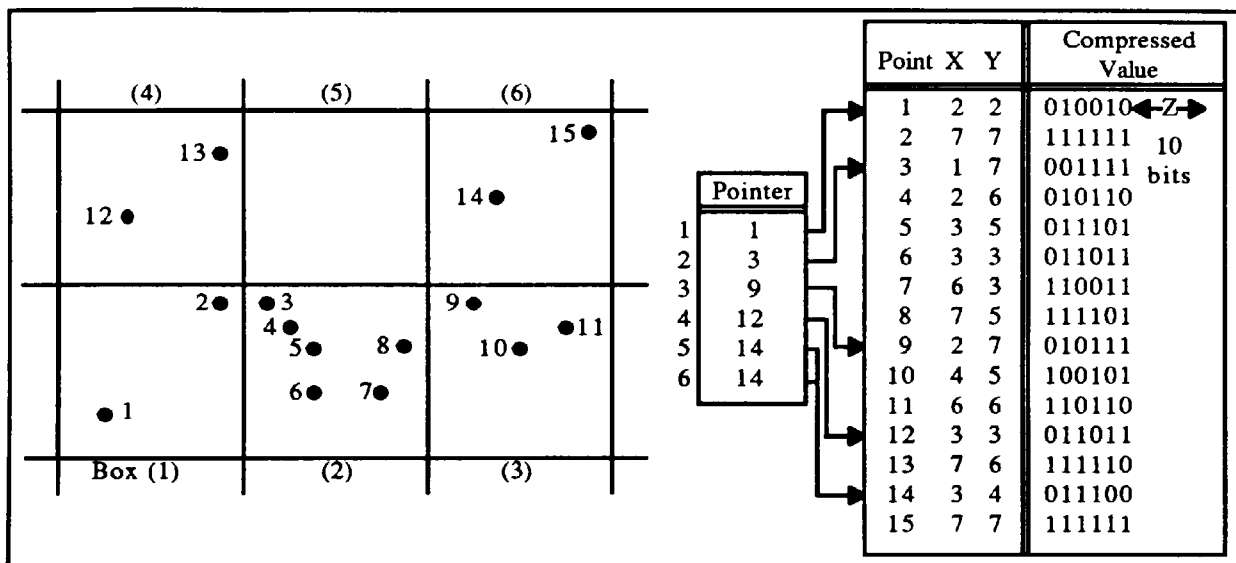


Figure 8.10 - Illustration of Coordinate Compression for TIN Vertices in Local Coordinates.

8.4 TIN Data Structures

The data structure of a triangulated irregular network can be represented in a number of different ways. Three main approaches have evolved in the generation of a subdivision of a surface into triangular facets. The primary entities of each represent the three primitive topological entities of a TIN - vertices, triangles and edges. A TIN data structure for encoding a triangulation can be thought of as the combination of these basic entities and a set of adjacency relations (De Floriani, 1987). Woo (1985) illustrates by virtue of an arrow diagram the nine possible relations that can be defined between pairs of primitive entities (Figure 8.11), where each arrow denotes an ordered relation between a pair of entities.

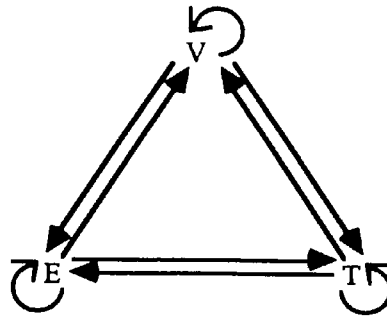


Figure 8.11 - Nine Possible Relations Between Pairs of Entities in a Triangular Grid.
(V: Vertices; E: Edges; T: Triangles).

De Floriani (1987) states that the topology of a triangular subdivision is completely and unambiguously represented by any suitably selected subset of these nine mutual adjacency relations. An illustration of these relations is presented below in Figure 8.12.

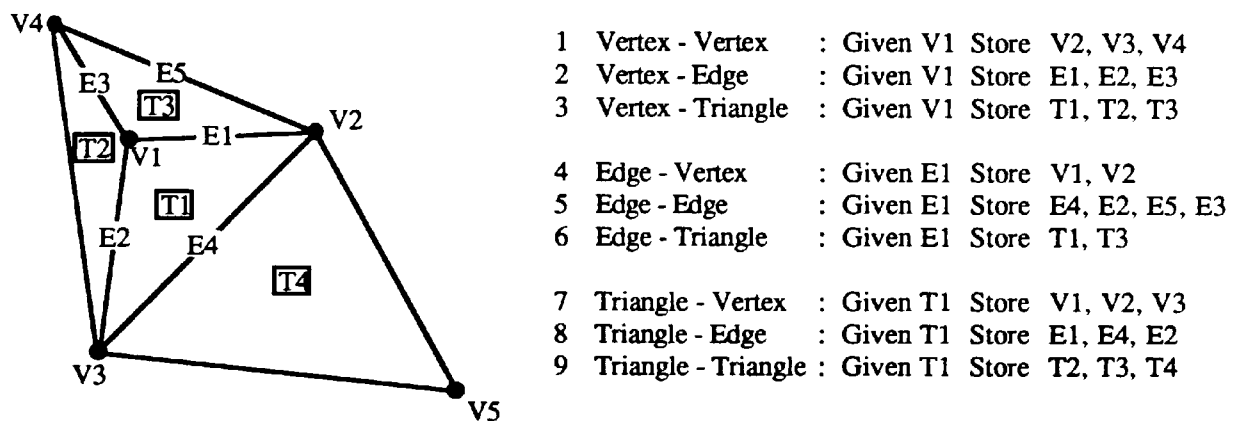


Figure 8.12 - Illustration of the Nine Possible Relations Between Pairs of Entities in a Triangular Grid (where V_n : Vertices; E_n : Edges; T_n : Triangles).

Vertex-Based TIN

In all TIN data structures, the x, y and z coordinates of every surface-specific point or vertex need to be defined, together with an index which uniquely references each point. In vertex-based TINs, such points are considered the primary entities of the data structure. Associated with each vertex is a list of pointer values indicating the position (and number) of connected points or edges, emanating from that vertex. With respect to the adjacency relationships, only the vertex-vertex relations and two entities (vertices and edges) are stored. The vertex-edge relations can be easily implied, whilst the vertex-triangle relations can be located by examining adjacent vertices. Since each edge is stored twice, storage may be reduced considerably by representing each edge only once, but a global search is required to find all the topological relationships.

Triangle-Based TIN

The second type of data structure regards the triangles as the primary entities, although the coordinates of the vertices still need to be stored in a secondary file. Each triangle is uniquely referenced and is defined by pointers to three corner points of the vertex-file, together with pointers to the three adjacent triangles. Thus, triangle-vertex and triangle-triangle relations are maintained within the data structure, whilst triangle-edge relations can be retrieved in constant time. This is the most popular form of representing a TIN, but it is not as storage-efficient as the vertex-based TIN. McKenna (1987) proposes the use of a hybrid data structure that utilises the list of connected points and the list of triangles, such that vertex-vertex and triangle-vertex relations are represented, whilst the vertex-edge and triangle-edge relations can be derived implicitly.

Edge-Based TIN

Heller (1990) advocates the use of an edge-oriented structure, since it is better suited for the swapping and splicing of triangle edges. De Floriani (1987) describes the modified winged-edge representation in which the three basic topological entities are stored together with the edge-vertex, edge-edge and edge-triangle relations. For each edge joining two vertices, its neighbouring triangular facets and two of its neighbouring edges are stored. Other relations may be derived efficiently, although the data structure can be considered to have the greatest storage overheads.

Each TIN data structure has evolved through the requirements of specific applications, such that they each have their own distinct advantages and disadvantages. The relationships that are incorporated within the TIN can be configured to the requirements of the application(s), such that a hybrid of topological relationships is allowable. However, the degree of TIN topology is directly related to storage costs. Factors which may further affect this depend upon the computer implementation of the data structure and whether the TIN supports static or dynamic triangulation. If a continuous update of the TIN is a prerequisite, such as for point insertion, the data structure will require sufficient flexibility to allow this. This may mean using linked or doubly-linked lists which place greater demands on storage space. Fixed-size lists or arrays can be more easily incorporated into static triangulation data structures, such that pointers are made redundant or can be replaced by indices to records. However, irrespective of these factors, it can be concluded that edge-based and triangle-based TINs are likely to require significantly more storage space than a vertex-based TIN.

From Euler's theorem, it can be shown that for a triangulation of N nodes, B of which are on the boundary, there are $2N-B-2$ triangles and a total of $3N-B-3$ distinct edges or $6N-2B-6$ pointers, if stored as edges from each vertex. For a vertex-based TIN derived from a regular grid DTM, the storage requirements may still be very high. For example, if 10% of original grid points are

needed to represent the surface satisfactorily, then the storage costs of the TIN could approach that of the original grid DEM. If the x , y and z coordinates together with the index correspond to $4N$ and the pointers to the connected points (edges) equal $6N-2B-6$ values, the storage costs are approximately $10N$ or 100% of the original grid storage. This assumes that the coordinates can be represented in the same storage space as the original grid elevation values and coordinate values are equivalent in storage to pointer values. The triangle-based TIN will require even greater storage, since for each of the $2N-B-2$ triangles, the three vertices and the three neighbouring triangles are stored ($12N-6B-12$), together with the vertex coordinates ($3N$). This corresponding storage requirement is equivalent to $15N$ or 150%, if 10% of the original grid points are used.

Whilst a TIN may be comprised of only a small proportion of original grid points, storage of the necessary topology to represent the spatial adjacency relations may incur too high a penalty. This is particularly true for large terrain databases. The explicit representation of adjacency relationships, such as edges in a vertex-based TIN, creates duplication of topology or data redundancy within the model. This can be illustrated by the fact that the 3N-B-3 edges are stored using 6N-2B-6 pointers.

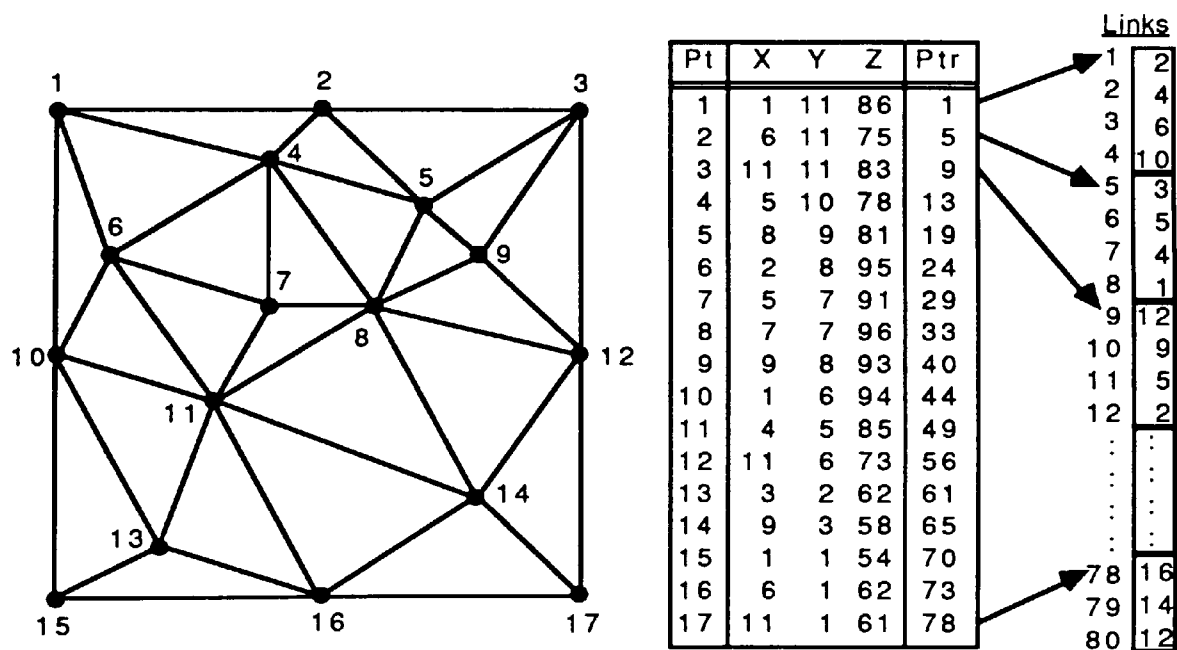


Figure 8.13 - Vertex-Based TIN Data Structure
($N=17$, $B=8$, $Edges=6N-2B-6=80$, $Triangles=2N-B-2=24$).

Despite this data duplication, the vertex-based TIN data structure (Peucker et al, 1978) is the representation which requires the least amount of storage capacity (Figure 8.13). However, the overhead of data duplication can be reduced to some extent by compressing the x, y and z coordinates of each vertex into two bytes of storage using the 'box-sort' addressing scheme (see

Section 8.3). The vertex-based TIN data structure itself is quite simple. Associated with each data point (vertex) is a list of Thiessen neighbours (ie. implicit edges), which are referenced by a pointer indicating the first of these neighbours. The number of edges originating from each vertex is implicitly derived from the pointer of the next vertex.

The storage requirements for the vertex-based TIN consist of storing for each point, its x,y and z coordinates, a pointer and its list of Thiessen neighbours. Since the vertices have been derived from a regular grid DEM, each of the coordinates can be stored as two-byte values, together with the pointers and links or neighbours. Hence,

$$\text{TIN STORAGE (x 2 bytes)} = (\text{No.of Points (N)} \times 4 (\text{x,y,z,pointer})) + \text{Total No. of Neighbours.}$$

As the number of points (N) within the TIN increases and the relative proportion of boundary points (B) decreases, the number of neighbours or edges (6N-2B-6) converges to 6N. Hence,

$$\text{TIN STORAGE} = \text{No.of Points} \times 10.$$

This illustrates the fact that the cost of explicitly storing the TIN data structure is approximately ten times greater than a regular grid for an equivalent number of nodes. Even if the vertex coordinates (x, y and z) are concatenated into two bytes of storage (as in Figure 8.10), the corresponding overhead ratio will still be 8:1. (For all of these storage costs, there is a small overhead in maintaining the 'box-sort' index).

8.5 TIN Terrain Results

TINs were constructed for ST06 and ST08 by iterative insertion of the points of largest error, until pre-specified tolerance levels were attained. At each level, the performance of the TIN was analysed with respect to storage cost, elevation errors and radio path loss errors. The number of elements within each TIN in terms of the three primary features (points or vertices, triangles and edges) are shown below in Table 8.1 (ST06) and Table 8.2 (ST08). The storage cost of each TIN as a percentage of original grid storage is given in the last column. This corresponds to the vertex-based TIN storage requirements of approximately 10 N, where N represents the number of original grid vertices. A simplified analysis of the TIN storage costs of Tables 8.1 & 8.2 is illustrated in Figure 8.14.

Max.Abs. Error	No. of Points	Boundary Points	% of Points	No. of Triangles	No. of Edges	% Storage
50 m	572	191	0.356	951	3044	3.316
45 m	585	191	0.364	977	3122	3.397
40 m	599	191	0.373	1005	3206	3.484
35 m	615	191	0.383	1037	3302	3.583
30 m	696	191	0.433	1199	3788	4.087
25 m	850	191	0.529	1507	4712	5.045
20 m	1087	191	0.676	1981	6134	6.519
15 m	1488	191	0.925	2783	8540	9.012
10 m	2440	192	1.517	4686	14250	14.931
5 m	4650	201	2.892	9097	27492	28.664

Table 8.1 - ST06 TIN Elements and Storage Costs Compared to 401x401 Regular Grid DEM.
(N.B. Triangles = $2N-B-2$, Edges = $6N-2B-6$, Storage = $[4N + \text{Edges}] \times 100\% / 401 \times 401$).

Max.Abs. Error	No. of Points	Boundary Points	% of Points	No. of Triangles	No. of Edges	% Storage
75 m	877	310	0.545	1442	4636	5.065
60 m	1010	311	0.628	1707	5432	5.891
55 m	1050	311	0.653	1787	5678	6.143
50 m	1117	313	0.695	1919	6070	6.553
45 m	1205	314	0.749	2094	6596	7.099
40 m	1350	314	0.840	2384	7466	8.001
35 m	1502	316	0.934	2686	8374	8.944
30 m	1794	317	1.116	3269	10124	10.759
25 m	2262	318	1.407	4204	12930	13.668
20 m	2923	326	1.818	5518	16880	17.769
15 m	4067	331	2.529	7801	23734	24.877
10 m	6837	355	4.252	13317	40306	42.073
5 m	13465	389	8.374	26539	80006	83.249

Table 8.2 - ST08 TIN Elements and Storage Costs Compared to 401x401 Regular Grid DEM.

The above tables and Figure 8.14 (overleaf) clearly show that the more constrained the TIN becomes (ie. the smaller the maximum absolute error), then storage costs increase sharply. The graph for ST08 shows that a TIN constrained to errors of ± 5 metres or less give no significant storage savings over the regular grid DEM. Whilst a TIN for ST08 can be constructed to a maximum absolute error tolerance of five metres using less than 8.5% of the original grid elevations, the cost of storing the necessary topology as a vertex-based TIN increases the storage overheads to over 83% of equivalent grid nodes. The corresponding storage cost for ST06

at five metres is less than 29%, using only 2.89% of original grid nodes. Based upon these results, it can be expected that the majority of British terrain (extracted from O.S. 50 metre regular grids) could be represented as TINs of maximum absolute error of ten metres, with corresponding storage savings of between 50 and 85%. The use of smaller tolerances will not give any significant saving in storage.

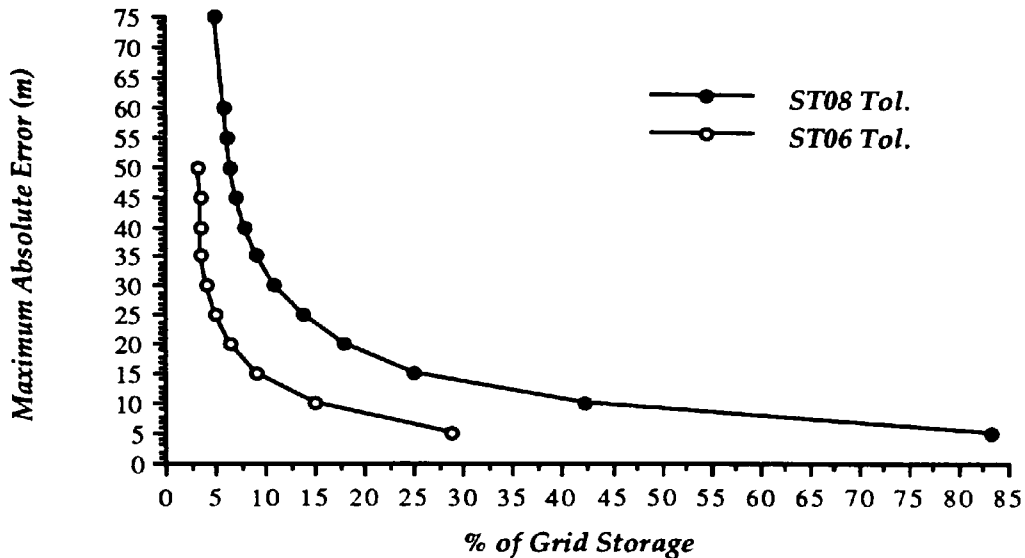


Figure 8.14 - TIN Storage for ST06 & ST08 with Respect to Maximum Absolute Error

The overall performance for surface representation of the TINs was analysed in terms of absolute average error, R.M.S. error, standard deviation and proportion of interpolated points within certain tolerances. These results are presented below in Tables 8.3 (ST06) and 8.4 (ST08). The graphs of storage against absolute average error and storage against R.M.S. error (Figure 8.15) display a similar relationship to the maximum absolute error in Figure 8.14 above.

Max.Abs. Error	% Storage	Abs.Ave. Error	RMSE	Stan. Dev.	% of Interpolated Points Within Error of				
					± 5 m	± 10 m	± 15 m	± 20 m	± 25 m
50 m	3.316	9.411	12.981	15.784	41.602	62.535	75.960	86.275	92.814
45 m	3.397	8.578	11.716	13.774	43.558	65.462	79.077	89.122	95.204
40 m	3.484	8.086	11.011	12.928	44.896	67.417	81.330	90.857	96.464
35 m	3.583	7.753	10.521	12.249	45.708	68.907	82.976	92.280	97.333
30 m	4.087	6.590	8.981	10.350	50.724	75.108	88.366	95.931	99.256
25 m	5.045	5.360	7.365	8.501	57.355	82.337	93.751	98.855	100.000
20 m	6.519	4.190	5.708	6.501	65.325	90.369	98.609	100.000	—
15 m	9.012	2.566	3.932	4.629	80.791	97.042	100.000	—	—
10 m	14.931	1.641	2.591	3.121	91.714	100.000	—	—	—
5 m	28.664	0.824	1.273	1.567	100.000	—	—	—	—

Table 8.3 - ST06 TIN Surface Elevation Performance.

Max.Abs. Error	% Storage	Abs.Ave. Error	RMSE	Stan. Dev.	% of Interpolated Points Within Error of				
					±5 m	±10 m	±15 m	±20 m	±25 m
75 m	5.065	17.307	22.175	29.766	20.991	38.080	52.662	64.650	74.296
60 m	5.891	14.511	18.491	23.414	24.424	43.903	59.497	71.779	81.251
55 m	6.143	13.323	17.066	22.003	26.568	47.368	63.416	75.679	84.697
50 m	6.553	12.424	15.868	20.620	28.122	49.810	66.220	78.556	87.402
45 m	7.099	11.309	14.451	18.338	30.635	53.480	70.214	82.420	90.618
40 m	8.001	9.815	12.502	15.844	34.159	59.117	76.247	87.942	94.969
35 m	8.944	8.757	11.191	14.387	37.938	64.170	80.940	91.447	97.234
30 m	10.759	7.282	9.435	11.952	45.044	72.666	87.598	95.478	99.096
25 m	13.668	6.038	7.791	9.974	51.635	80.097	93.399	98.799	100.000
20 m	17.769	4.837	6.221	7.851	60.322	88.285	98.106	100.000	—
15 m	24.877	3.710	4.775	6.152	71.617	95.626	100.000	—	—
10 m	42.073	2.540	3.274	4.186	86.826	100.000	—	—	—
5 m	83.249	0.938	1.492	1.892	100.000	—	—	—	—

Table 8.4 - ST08 TIN Surface Elevation Performance.

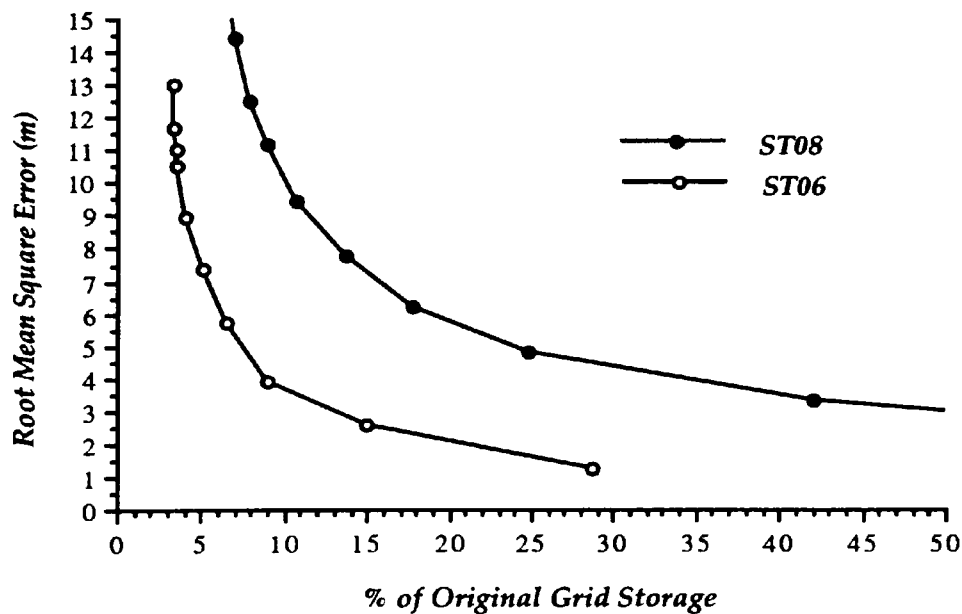


Figure 8.15 - Elevation RMS Error v. Storage for ST06 & ST08.

The TINs at a tolerance of ± 10 metres maximum error are illustrated overleaf in Figure 8.16 (ST06) and Figure 8.18 (ST08), together with their corresponding contour maps (Figures 8.17 & 8.19), which have been linearly interpolated. A shaded relief map of the ST08 ten metre TIN corresponding to Figures 8.18 & 8.19 is also presented in Figure 8.21. This can be compared to the shaded relief map of the original surface (Figure 8.20).

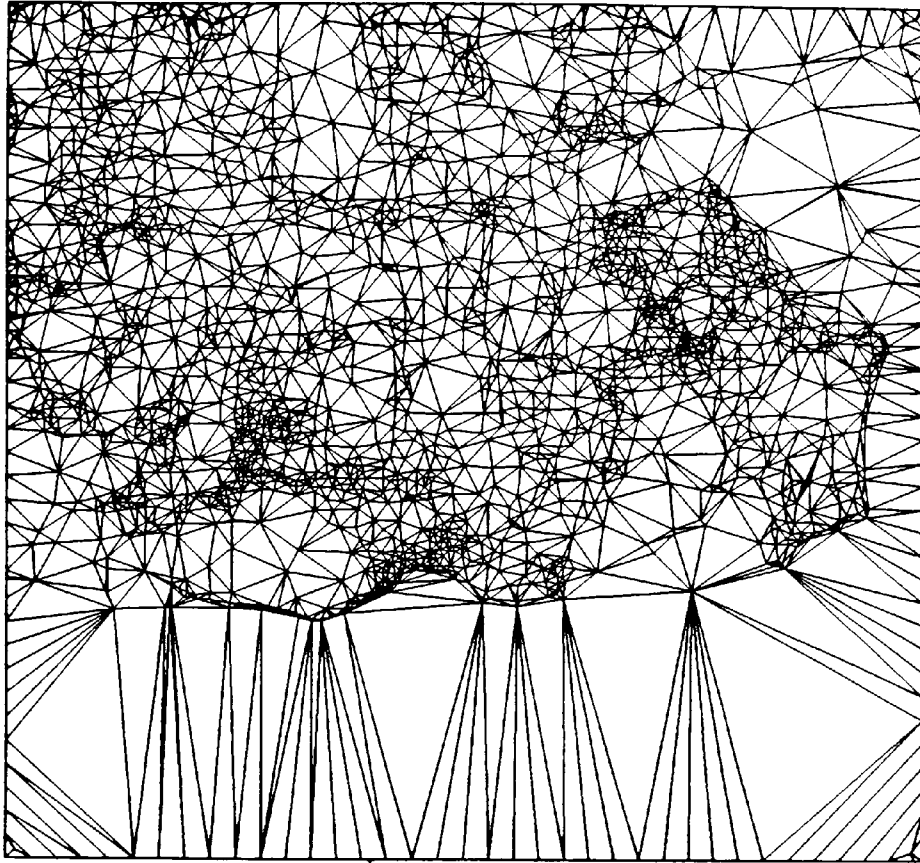


Figure 8.16 - ST06 TIN Surface at a Maximum Absolute Error Tolerance of 10 Metres.

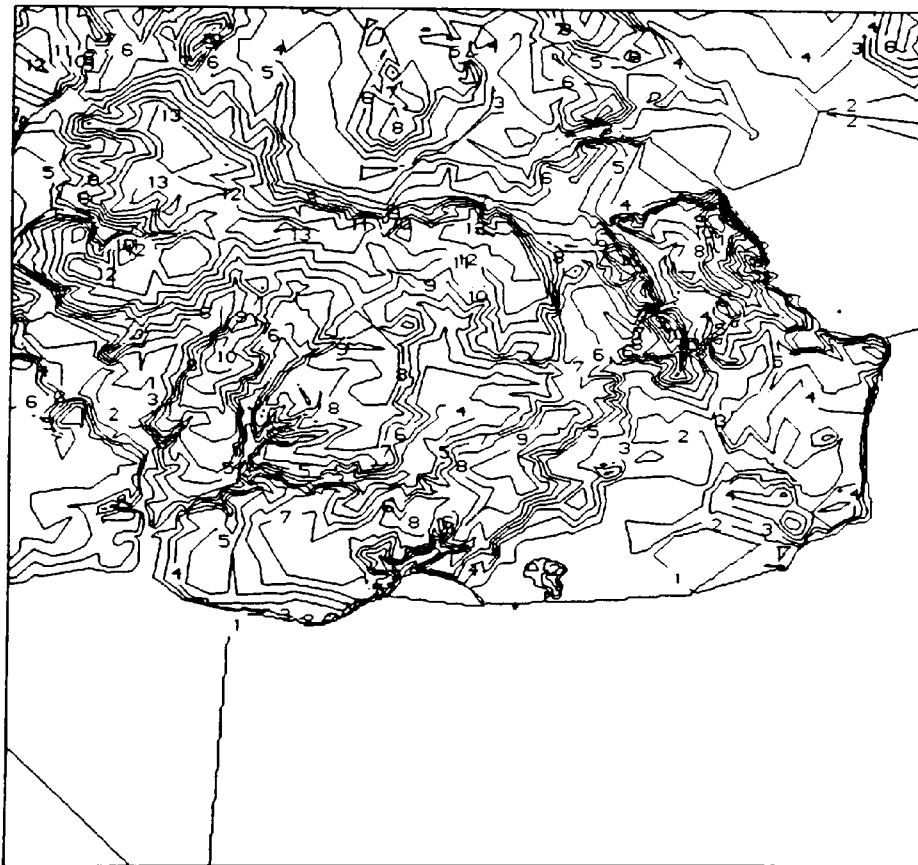


Figure 8.17 - ST06 Linearly Interpolated Contour Surface From the 10 Metre TIN of Figure 8.16.

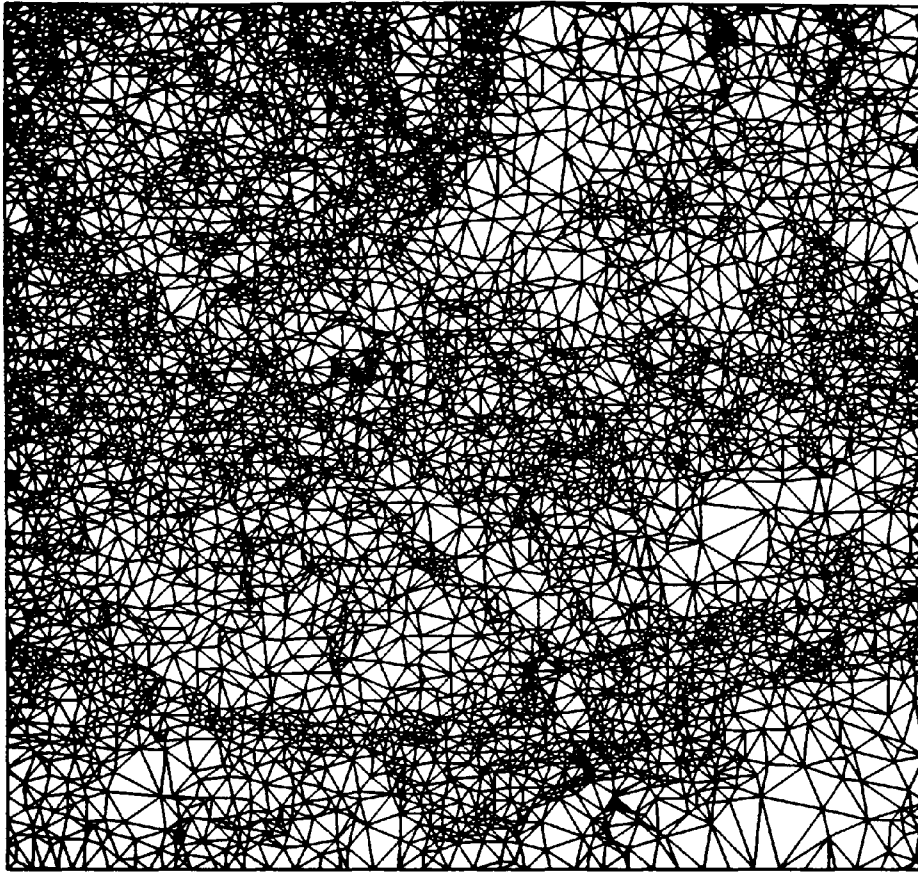


Figure 8.18 - ST08 TIN Surface at a Maximum Absolute Error Tolerance of 10 Metres.

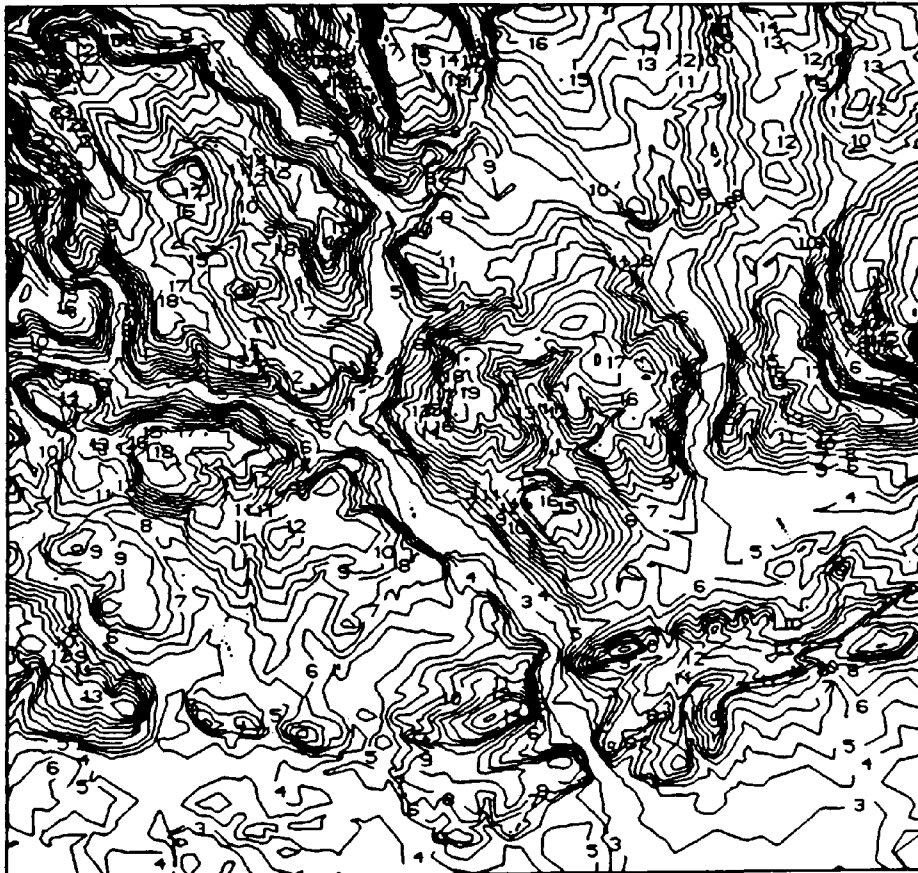


Figure 8.19 - ST08 Linearly Interpolated Contour Surface From the 10 Metre TIN of Figure 8.18.



Figure 8.20 - ST08 Shaded Relief Map of the Original Grid DEM.



*Figure 8.21 - ST08 TIN Shaded Relief Map at a Maximum Absolute Error Tolerance of 10 Metres.
(N.B. This surface has been interpolated using 4.25% of the Grid Points of Figure 8.20 above).*

It is apparent from both TIN surfaces (Figures 8.16 & 8.18), that there is a higher concentration of triangles in the areas of greatest terrain variability, such as valleys and mountains, whilst fewer vertices are incorporated in less variable regions, or flatter areas. This clearly proves the adaptable nature of the triangulated irregular network as a DEM which contains very little redundancy, in terms of data points. The TIN shaded relief map of ST08 (Figure 8.21) further illustrates this adaptability. A visual comparison with the original surface (Figure 8.20) shows that all the critical terrain features are incorporated. However, the triangular origins of the surface are clearly visible in less variable regions.

The contour maps of the corresponding TINs for ST06 & ST08 (Figures 8.17 & 8.19) illustrate the linear nature of the surfaces, particularly in the regions of sparser triangles. Even though the TIN surface is continuous, some contours appear jagged and have a tendency to change sharply or abruptly. Also, on closer examination and in particular for TINs of lower tolerances, the contours do not cluster as tightly in areas of steep slope. Another anomaly which arises in Figure 8.17 is the contour labelled '1' at sea level. Whilst this appears unsightly, the offending, slightly sloping triangles contribute an insignificant degree of error to the overall model. Such occurrences for ST06 are caused by triangle edges linking vertices of very low elevation (one or two metres) with a sea level vertex. This is exaggerated in Figure 8.17, since the edges concerned are in excess of five kilometres. The problem can be easily overcome by the insertion of sea level vertices near the coastline, or inserting all the points along the coastline. However, since contouring is not the major application of these TINs, this was not considered further.

Indeed, for applications such as contouring, the linear nature of the TIN makes it unsuitable unless contour smoothing is incorporated. This can be accomplished using surface patches which include derivative information, such that the elevations at the TIN vertices are maintained, together with the continuity along the triangle edges (McCullagh, 1979). McCullagh (1981) describes an algorithm to accomplish this and compares its interpolation performance with rectangular gridded methods. Gold & Cormack (1987) also describe the generation of contour maps from a TIN using a Delaunay triangulation algorithm.

8.6 TIN Profile Interpolation

The algorithm for TIN profile interpolation incorporates a line following approach through the network of triangles. Points are linearly interpolated at the intersection of the profile with the triangle edges. The complexity of this algorithm depends upon the nature of the TIN data structure and the topological relationships which are maintained. Hence, the most efficient data structure is one in which the triangles are the primary entities, since the path of

the profile can be easily tracked from one triangle to the next. This can be accomplished by checking each of the three neighbouring triangles at each stage. However, the most storage efficient TIN is the vertex-based model discussed and implemented in Section 8.4. As such, profile interpolation is not as efficient, since the triangle edges through which the profile intersects have to be searched from the list of Thiessen neighbours of each vertex.

Regardless of TIN data structure, the algorithm for interpolation will initially need to search for the triangle that encloses the start or end-point of the profile. For a vertex-based TIN, a useful premise is to assume that the vertex closest to the end-point will form one of the vertices of the enclosing triangle. This will more often than not be the case, but in certain circumstances this premise is found to be false. One occurrence of this is when four vertices are 'nearly' co-circular and the interpolant is closer to the vertex outside the circumcircle of the valid Delaunay triangle. This is illustrated in Figure 8.22 below.

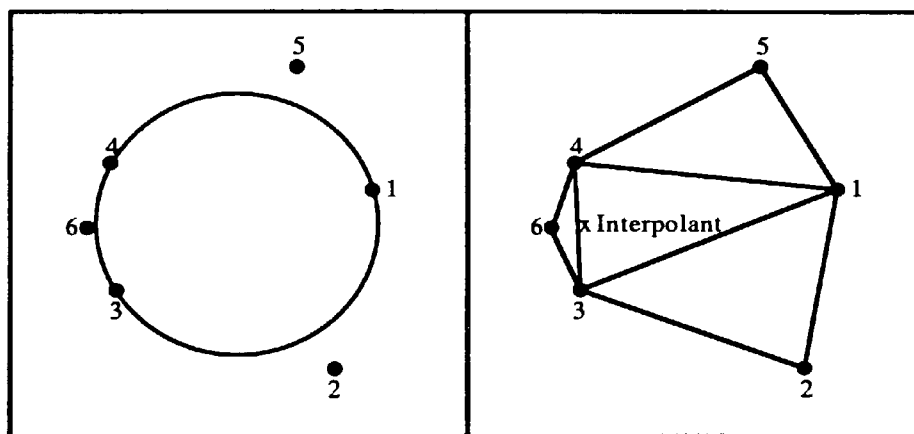


Figure 8.22 - Illustration of the Nearest Vertex (6) to the Interpolant (x) Lying Outside the Triangle Which Includes That Vertex.

Most algorithms for locating the nearest vertex start from an arbitrary vertex and move to the Thiessen neighbour which is closest to the interpolant, until no more traverses can be made. However, this can be accomplished more efficiently if the box-sort data structure is used to index the points, since the nearest vertex can be located from sampling only the vertices in the local neighbourhood of the interpolant. For each two consecutive neighbours of this vertex, a check can be made to determine whether the interpolant lies within this bounding triangle. If no bounding triangle is found the search is repeated for the next closest vertex to the interpolant. The test for whether a point lies within a triangle is illustrated overleaf in Figure 8.23.

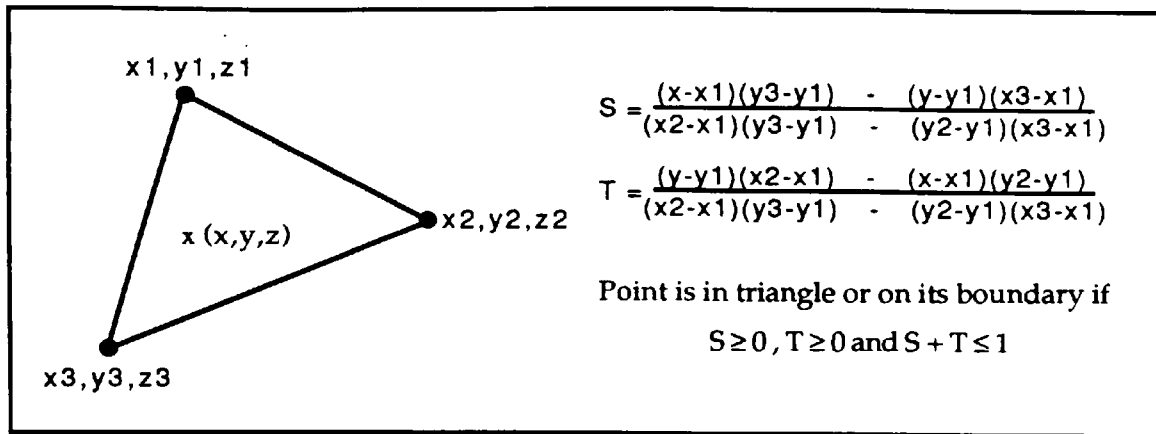


Figure 8.23 - Test for Determining Whether a Point (x,y,z) Lies Within A Triangle.

The point within a triangle can then interpolated as:

$$z = \frac{d - ax + by}{c}$$

$$\text{where } a = \begin{vmatrix} y_1 & z_1 & 1 \\ y_2 & z_2 & 1 \\ y_3 & z_3 & 1 \end{vmatrix} \quad b = \begin{vmatrix} x_1 & z_1 & 1 \\ x_2 & z_2 & 1 \\ x_3 & z_3 & 1 \end{vmatrix} \quad c = \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} \quad d = \begin{vmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{vmatrix}$$

Interpolation continues by examining each of the triangle sides to determine which one the profile intersects. The third point of the next intersected triangle is found as the common neighbour of both vertices of the previously intersected edge. Hence, interpolation through the network of triangles consists of searching for the common neighbour of an edge and determining the correct intersection at each stage, until the profile end-point is found.

The number of profile interpolations is equivalent to the number of edge intersections plus two, so is directly related to how the TIN surface adapts to the variability of the original surface. Grid interpolation is carried out at regular intervals and is not adaptive to the terrain, since the grid sampling interval is constant. However, in the TIN all the critical points of the profile will be identified regardless of the number of points in the profile. These critical points at edge intersections correspond to the changes in slope along the profile, so no redundant information is interpolated. This is illustrated overleaf in Figure 8.24.

The number of profile interpolations and hence, profile interpolation time is directly related to the number of edges, triangles and vertices within the TIN. However, for individual profiles the time taken to interpolate two profiles of similar distance can vary considerably, depending upon the nature and variability of the terrain. Profile interpolation time is therefore directly related to the number of TIN vertices. This is shown below in Figure 8.25 for error constrained TINs at maximum errors of ± 5 to ± 20 metres.

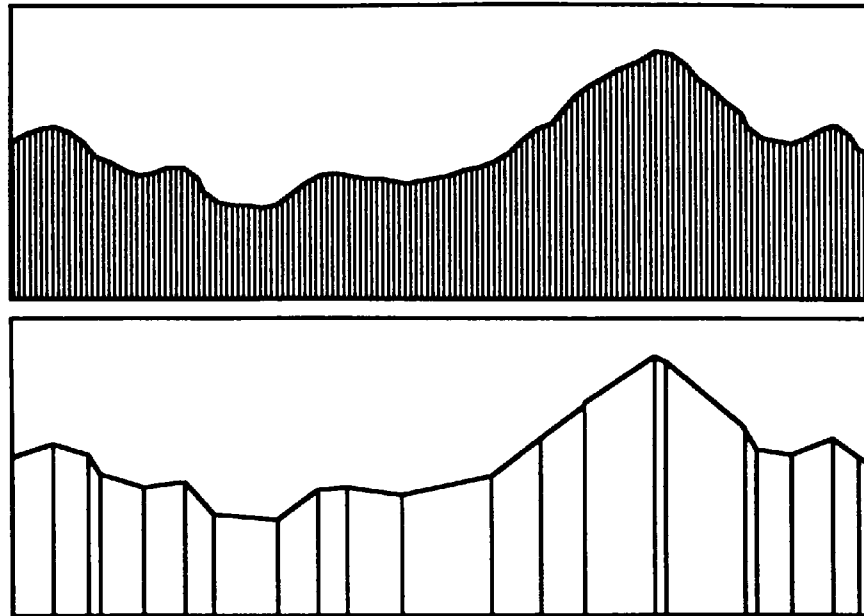


Figure 8.24 - Grid Interpolation (Constant Intervals) and TIN Interpolation at Triangle Intersections (Change in Slopes).

Hence average interpolation time is 12 milliseconds for the densest TIN (ie. ST08 at ± 5 metres). This corresponds to a time of approximately 40 times slower than grid interpolation in an error-free 50 metre grid. This profile interpolation algorithm is not optimal, since tests using a triangle-based TIN show that interpolation time can be improved by a factor of two or three. However, this is achieved at a cost of increasing storage. In general therefore, profile interpolation in a TIN will never be as efficient as linear or bilinear interpolation within a regular grid DEM, due primarily to the need to search the data structure for TIN intersections. Therefore, the overheads of TIN interpolation can only be tolerated if the storage savings of such a data structure are substantial. This is considered further, later in this chapter.

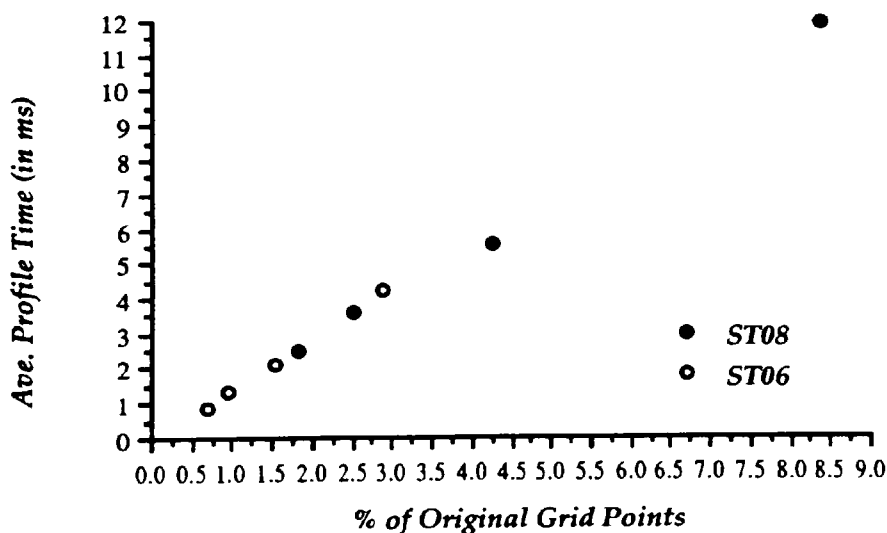


Figure 8.25 - Ave. Profile Interpolation Time (in msec) in Relation to Number of TIN Vertices. (This is shown for the ± 5 to ± 20 metre Maximum Error (Constrained) TINs of ST06 & ST08).

8.7 Radio Path Loss Results

The algorithm for calculating radio path loss accepts both regularly and irregularly sampled profile elevation data as input. The results of this algorithm are shown below in Table 8.5 (ST06) and Table 8.6 (ST08) for each of the error tolerant TINs described in Section 8.5. As for the other DEMs, these results represent the average of 1000 randomly generated test profiles.

Abs. Max. Error	Frequency = 200 MHz				Frequency = 400 MHz				Frequency = 600 MHz			
	A.Av. Error	RMSE	No. of Profs ≤ 6dB 12dB		A.Av. Error	RMSE	No. of Profs ≤ 6dB 12dB		A.Av. Error	RMSE	No. of Profs ≤ 6dB 12dB	
50 m	5.991	7.663	578	875	7.164	9.431	517	797	7.583	10.122	486	768
45 m	5.454	7.141	633	899	6.495	8.789	562	827	6.899	9.407	536	796
40 m	5.294	6.967	651	908	6.246	8.518	595	837	6.632	9.090	566	806
35 m	5.135	6.855	654	912	6.008	8.273	609	845	6.410	8.817	573	823
30 m	4.444	6.003	754	943	4.887	6.925	701	898	5.131	7.197	659	893
25 m	3.779	5.149	795	970	4.001	5.771	767	943	3.990	5.751	766	940
20 m	3.225	4.558	843	982	3.284	4.821	814	970	3.147	4.632	824	975
15 m	2.638	3.868	898	986	2.659	4.086	874	981	2.533	3.862	871	987
10 m	1.996	3.145	934	990	1.841	3.043	922	994	1.722	2.853	931	999
5 m	1.136	2.065	976	997	1.014	2.016	971	997	0.978	1.912	980	1000

Table 8.5a - ST06 Radio Path Loss Errors for TINs at 200, 400 & 600 MHz.

Abs. Max. Error	Frequency = 900 MHz				Frequency = 1400 MHz				Frequency = 1800 MHz			
	A.Av. Error	RMSE	No. of Profs ≤ 6dB 12dB		A.Av. Error	RMSE	No. of Profs ≤ 6dB 12dB		A.Av. Error	RMSE	No. of Profs ≤ 6dB 12dB	
50 m	9.095	12.048	422	700	10.884	14.264	366	618	11.730	15.371	350	566
45 m	8.179	11.102	483	730	9.654	13.036	442	660	10.376	14.014	430	621
40 m	7.836	10.718	492	750	9.226	12.564	455	679	9.917	13.511	442	642
35 m	7.650	10.433	495	758	9.036	12.270	450	688	9.724	13.220	437	647
30 m	5.922	8.316	589	840	6.782	9.584	555	790	7.258	10.305	538	761
25 m	4.499	6.564	696	908	5.090	7.532	674	870	5.439	8.104	662	843
20 m	3.478	5.171	789	956	3.894	5.877	771	927	4.138	6.306	751	906
15 m	2.728	4.261	845	980	3.047	4.842	820	958	3.226	5.185	809	948
10 m	1.866	3.172	905	994	2.070	3.575	893	986	2.189	3.820	886	978
5 m	1.112	2.104	967	997	1.174	2.224	960	995	1.286	2.346	953	991

Table 8.5b - ST06 Radio Path Loss Errors for TINs at 900, 1400 & 1800 MHz.

Abs. Max. Error	Frequency = 200 MHz				Frequency = 400 MHz				Frequency = 600 MHz			
	A.Av. Error	RMSE	No. of Profs ≤ 6dB 12dB		A.Av. Error	RMSE	No. of Profs ≤ 6dB 12dB		A.Av. Error	RMSE	No. of Profs ≤ 6dB 12dB	
75 m	7.457	10.200	531	798	7.891	10.933	510	768	7.905	10.955	520	760
60 m	6.456	9.005	603	832	6.603	9.316	585	820	6.610	9.280	592	808
55 m	6.028	8.442	633	856	6.187	8.783	604	841	6.121	8.693	625	831
50 m	5.987	8.442	638	870	6.169	8.790	608	848	6.030	8.643	630	836
45 m	5.648	7.992	656	886	5.617	8.106	640	871	5.548	8.001	654	860
40 m	5.168	7.346	688	911	5.046	7.253	664	909	4.970	7.245	682	890
35 m	4.986	7.007	691	911	4.845	7.028	691	913	4.735	6.985	710	903
30 m	4.601	6.561	742	925	4.354	6.557	741	922	4.189	6.368	751	926
25 m	4.001	6.037	792	939	3.733	5.857	783	942	3.577	5.686	801	941
20 m	3.519	5.448	831	957	3.249	5.187	813	962	3.049	5.040	826	963
15 m	2.926	4.934	869	966	2.633	4.495	853	974	2.377	4.255	874	978
10 m	2.186	3.597	922	986	1.876	3.127	914	993	1.757	3.164	915	989
5 m	1.201	2.061	978	996	1.013	1.948	976	997	0.965	1.817	975	997

Table 8.6a - ST08 Radio Path Loss Errors for TINs at 200, 400 & 600 MHz.

Abs. Max. Error	Frequency = 900 MHz				Frequency = 1400 MHz				Frequency = 1800 MHz			
	A.Av. Error	RMSE	No. of Profs ≤ 6dB 12dB		A.Av. Error	RMSE	No. of Profs ≤ 6dB 12dB		A.Av. Error	RMSE	No. of Profs ≤ 6dB 12dB	
75 m	8.329	11.514	504	732	9.179	12.747	486	701	9.692	13.494	466	672
60 m	6.902	9.679	572	789	7.574	10.718	553	767	7.994	11.364	535	754
55 m	6.380	9.047	601	815	6.989	9.997	583	787	7.382	10.609	562	773
50 m	6.223	8.952	613	825	6.803	9.881	596	796	7.190	10.493	583	782
45 m	5.790	8.397	648	848	6.338	9.290	634	818	6.702	9.874	618	800
40 m	5.202	7.566	670	875	5.689	8.379	651	852	6.013	8.907	639	831
35 m	4.981	7.353	693	886	5.446	8.162	675	863	5.756	8.697	661	847
30 m	4.383	6.722	741	904	4.782	7.379	721	884	5.054	7.850	704	869
25 m	3.670	5.876	789	934	4.004	6.532	774	917	4.231	6.972	766	909
20 m	3.172	5.277	816	952	3.419	5.823	805	936	3.603	6.194	797	930
15 m	2.401	4.398	876	972	2.577	4.906	868	958	2.719	5.220	866	953
10 m	1.800	3.287	910	986	1.924	3.636	905	978	2.035	3.879	902	975
5 m	0.998	2.005	970	993	1.119	2.306	965	991	1.204	2.569	961	990

Table 8.6b - ST08 Radio Path Loss Errors for TINs at 900, 1400 & 1800 MHz.

The graphs of average radio path loss R.M.S. errors for ST06 and ST08 are presented overleaf in Figure 8.26 and the average number of profiles within errors of ± 6 dBs and ± 12 dBs in Figure

8.27. Both of these graphs represent the results averaged over the radio frequencies of 200 to 1800 MHz.

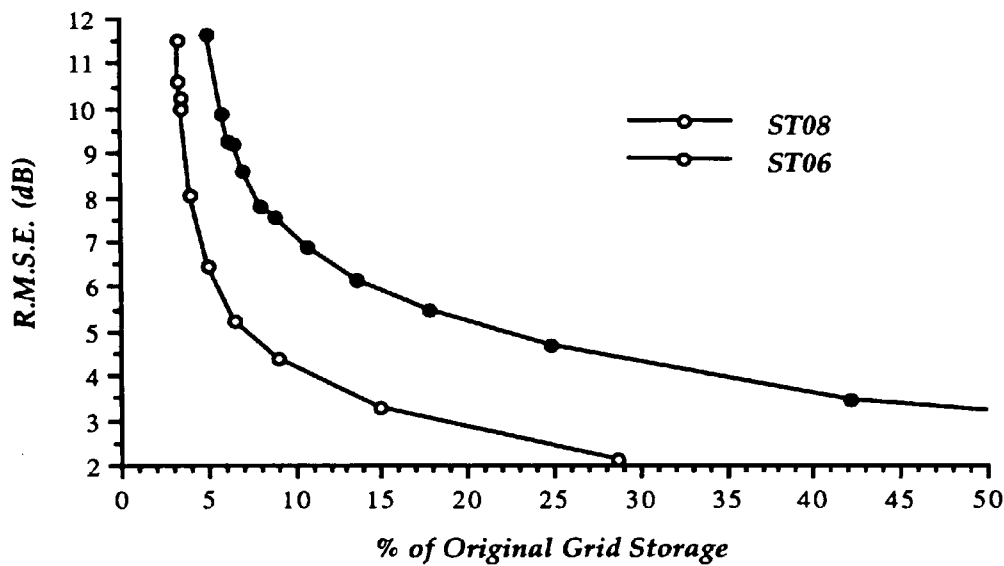


Figure 8.26 - R.M.S. Error Radio Path Losses for ST06 & ST08
Averaged Over the Frequencies of 200, 400, 600, 900, 1400 & 1800 MHz.

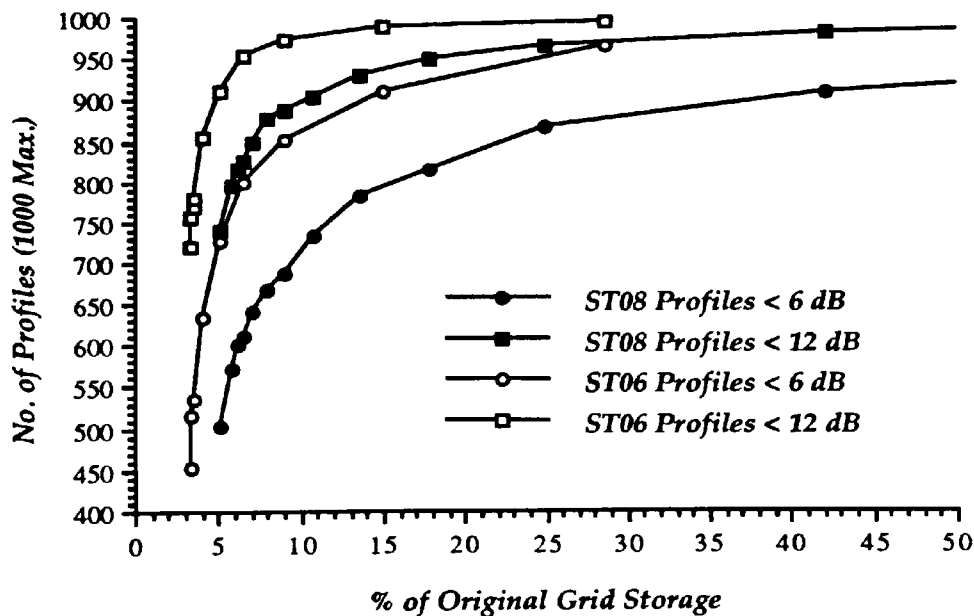


Figure 8.27 - Number of TIN Profiles Within Radio Path Loss of 6 and 12 dBs for ST06 & ST08
Averaged Over the Frequencies of 200, 400, 600, 900, 1400 & 1800 MHz.

The behaviour of these graphs is very similar to those for the polynomial DEMs and the surface patch quadtree DEM. However, the storage costs for the TIN are much greater than these other DEMs for comparable radio path loss performance.

8.8 TIN Conclusions

The triangulated irregular network is the most popular alternative to the regular grid as a digital terrain model. The TIN overcomes the problem of data (vertex) redundancy, since it adapts itself to the variability of the terrain. The TINs of Figures 8.16 & 8.18 illustrate the concentration of vertices in regions of high variability. Hence every vertex is a critical point which contributes to the fidelity of the surface model. As such, the number of original grid points used to represent the surface as a TIN is greatly reduced. The results have shown for ST06 & ST08 that less than 8.5% of original grid points (sampled every 50 metres) are needed to contribute to a surface which has a maximum absolute error of five metres. Since the TIN is adaptable to terrain variability, it is also error tolerant, so the user can specify constraints on its construction, such that maximum or average error is limited to pre-specified tolerances.

Much research has been undertaken to identify critical terrain points from a regular grid DEM for use in a TIN. However, of the published algorithms there appears to be no such method of selecting TIN vertices which provide a 'satisfactory' surface model. Automated one or two pass point selection methods are too dependent upon the subjective selection criteria of the user. As such, redundant points need to be further eliminated and missing critical points added, whilst the overall fidelity of the TIN with respect to maximum or absolute error cannot be constrained. The future of TIN vertex selection techniques appears to rest with iterative procedures which successively add or delete grid vertices until a pre-defined error tolerance is satisfied.

The algorithm used to construct the triangulation is an implementation of the Delaunay triangulation. It has the properties of stability, equilateralness and non-intersection of triangles, thus producing a unique triangulation (McCullagh, 1987). However, the use of TIN vertices derived from a regular grid creates an anomaly which threatens the uniqueness of this triangulation. At present, current Delaunay triangulation algorithms do not consider the case of degeneracies arising, even though this is very probable if the vertices are derived from a grid. As national mapping organisations continue the rapid growth in supplying grid DEMs, it is to be expected that with the parallel growth in the application of TINs, the Delaunay triangulation will receive more attention with regard to degeneracies. A further problem with the Delaunay triangulation is its unconstrained nature during construction. The fidelity of ridge and channel lines cannot be guaranteed in the final model, since two points may not necessarily be connected by a triangle edge. Instead the ridge or channel may be cut by an edge connecting two other vertices. Such a problem can be crucial for applications such as profiling, visualisation or calculating radio path loss, where the importance of maintaining such critical features is paramount. It is important therefore that if such data is available, the Delaunay triangulation should be constrained to fit these surface features. McCullagh (1983), Kettelman (1987), De Floriani & Puppo (1988) and Scarlatos & Pavlidis (1991) have all identified this

problem and suggest methods or algorithms for constraining Delaunay triangulations.

It has been shown that the degree of surface variability is an important factor in the number of TIN vertices employed. Whilst this will generally be less than 10% of original grid points (sampled at 50 metre intervals), the actual TIN storage savings may be no better than 0 to 20%. The explicit storage of TIN topology, primarily due to pointer maintenance in defining the adjacency relationships of the triangulation, adds greatly to data volume thus counteracting the benefits of a greatly reduced volume of vertices. Whilst the TIN removes the data point redundancy of grid DEMs, the explicit representation of adjacency relationships, such as edges in a vertex-based TIN, creates duplication of topology or another form of data redundancy (Kidner & Jones, 1991). This can be illustrated by the fact that the $3N-3$ edges of the TIN are stored using $6N-6$ pointers. A storage-efficient alternative to this problem is to store only the triangle vertices and to reconstruct the surface locally in an area of interest, using the Delaunay triangulation. This approach is termed 'implicit triangulation' (Kidner & Jones, 1991).

8.9 Implicit Triangulation

The Implicit TIN significantly reduces the storage overheads of conventional (explicit) TINs by eliminating the need for permanent representation of the adjacency relationships. Assuming that a typical node valency is six, and the storage space of a pointer (Thiessen neighbour) is one third of a vertex, the space requirements of a triangulation can be reduced by at least two thirds. However, the use of a spatial addressing mechanism, such as the 'box-sort' data structure, not only allows an efficient search, but provides a method of compressing the vertex coordinates (Figure 8.10). As such, storage savings of 90% can be attained, when compared to a conventional vertex-based TIN. Subsequent use of the Delaunay triangulation provides the necessary consistency of reconstruction to retrieve the local surface at a pre-specified tolerance (Kidner & Jones, 1991). Hence the Delaunay triangulation procedure is considered an integral component of the TIN.

Since the Implicit TIN minimises storage, the topological relations have to be searched or derived algorithmically instead of retrieved directly. Hence, storage efficiency is increased at a cost of decreasing computational efficiency. However, the Implicit TIN is particularly suited to applications which only require access to a subset of the full terrain model at any one time, such as in profiling, visibility analysis, earthwork calculations, communication network siting or site planning etc. In such instances, conventional explicit DEMs maintain the full surface topology, even though only a small subset may only be accessed at any one time.

The storage overhead of an explicit vertex-based TIN is equivalent to ten times the number of selected vertices and even greater for a triangle or edge-based TIN. It is therefore feasible to assume that for some TINs, the storage costs may be greater than the original grid DEM, from which it may have been derived. With the Implicit TIN, this problem will never arise, since the cost of storing each vertex in the TIN is two bytes of storage, which is equivalent to the cost of storing a regular grid elevation. In the worst instance, when the x, y and z coordinates are not concatenated, storage is only three times greater than each selected grid vertex. Some other, alternative approaches to overcoming the data handling problem for large terrain databases have focused upon either multiple resolution TINs, in which a pyramid-structure is used to represent the surface at different levels of resolution, or alternatively hierarchical TINs, in which different parts of the surface are represented at different levels of resolution (Samet, 1990; De Floriani, 1987, 1989).

The Implicit TIN is constructed in exactly the same way as the explicit TIN, with an iterative procedure utilising a dynamic data structure. However, once the triangulation is complete, all the topological relationships defining the TIN are discarded. At the application stage, efficient spatial search of the data can restrict the retrieval of points to discrete windows of interest. At this local level, the subsequent use of the Delaunay triangulation will regenerate the original TIN surface, whilst maintaining the integrity of the original TIN within this region. An important prerequisite of the implicit TIN is an efficient spatial search of vertices, which can be accomplished with the box-sort structure outlined in Section 8.3. All TINs have to perform searches for some topological relationships, but the implicit TIN expands this search to derive all the necessary topological relationships. Only the triangle vertices are stored, so all applications will require the derivation of edges or triangles as required using a Delaunay triangulation algorithm.

An important characteristic of the Delaunay triangulation is its uniqueness for a given set of non-degenerate points. When triangulating a subset of the database of vertices however, the triangulation at the boundary of the subset (ie. the convex hull) cannot be guaranteed to be equivalent to the original global triangulation, since neighbouring vertices outside the region of interest will not be taken into account. To ensure that this does not have any major effect upon the required application, the spatial search of vertices for triangulation should allow a 'reasonable' margin of points around the application window. This margin should be dependent upon the sparseness of the data within the terrain model. For any spatial window, surround conditions can be set such that the search for vertices extends outwards until a minimum of points is obtained which ensures the accuracy of the reconstruction will not be affected by the boundary.

The use of the Implicit TIN can be illustrated for profile interpolation through a surface. The

grid cells covering the region of interest, together with a 'sufficient' external boundary can be directly accessed using the box-sort data structure to determine all the vertices of the local TIN. The convex hull of this subset is calculated for defining the boundary conditions of the TIN. The implicit topology is retrieved by the application of the Delaunay triangulation algorithm, such that the geometrical properties of the original TIN are unchanged. This is illustrated below in Figure 8.28, for a profile interpolated from the implicit TIN for ST08, constrained to a maximum absolute error of ± 5 metres.

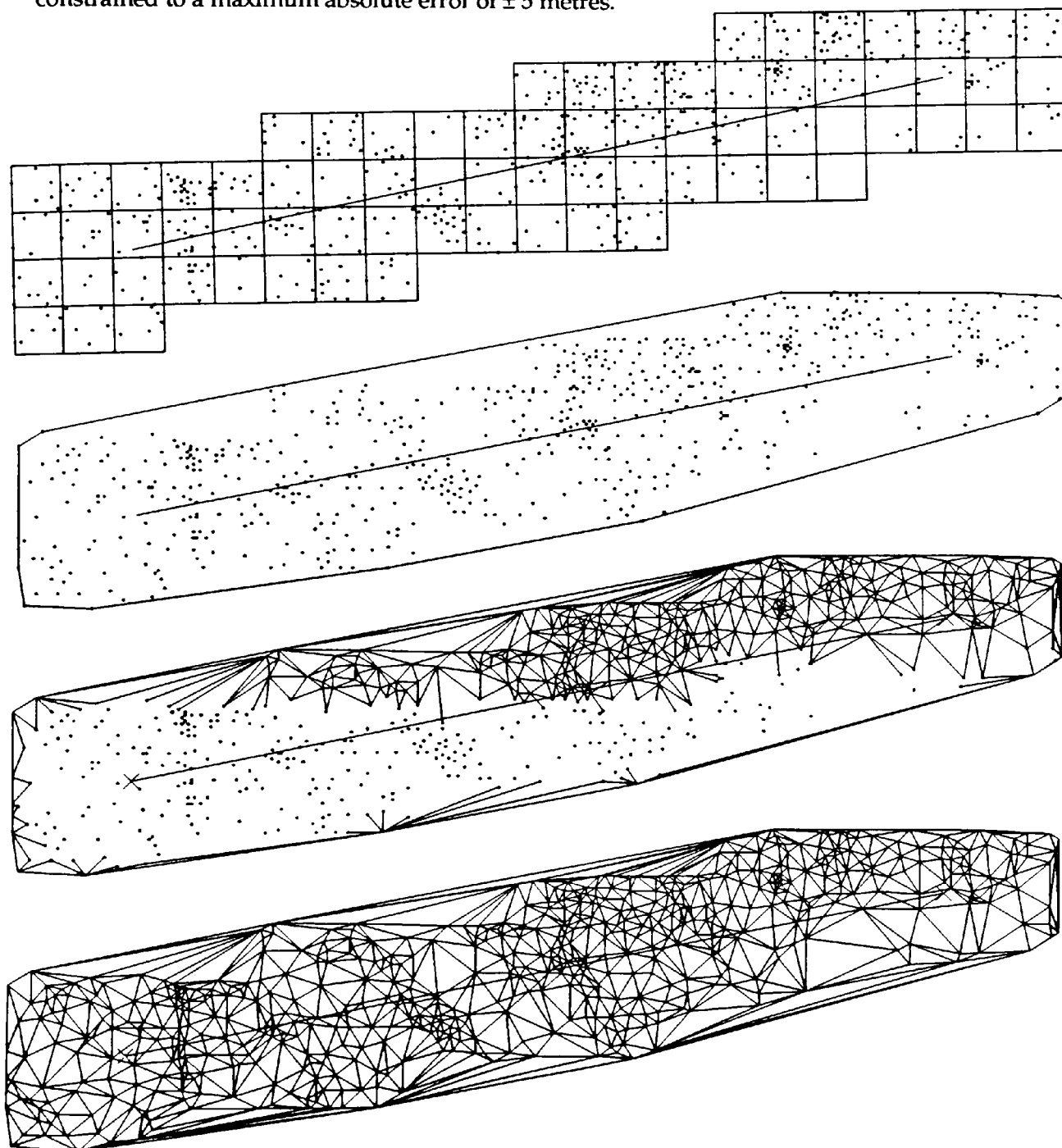


Figure 8.28 - Profile Interpolation Within the Implicit TIN.

- (i) Determine all the vertices of the local TIN;*
- (ii) Construct the Convex Hull;*
- (iii) Retrieve the Original TIN Topology by Local Re-Triangulation;*
- (iv) Reconstructed Profile TIN.*

The Implicit TIN supports the use of a multiple resolution DEM for surface modelling. A box-sort data structure can be maintained for each resolution of the DEM, such that TINs can be modelled at resolutions or error tolerances of ± 50 metres to ± 5 metres with no data redundancy or data duplication. This can be considered a form of the Delaunay Pyramid (De Floriani, 1989), but without the need to represent the TIN topology or the hierarchical links between triangles at different resolutions. Therefore, to construct a local TIN for profile interpolation, the TIN vertices at each level would be directly retrieved using the box-sort data structure and merged into a common data set for triangulating. This allows the user greater flexibility, since the terrain model is not constrained to one predetermined error tolerance. This is illustrated overleaf (Figure 8.29) for a profile through ST08 at maximum absolute error tolerances of 20 metres to 5 metres.

The most significant advantage of the Implicit TIN is its immense storage savings over conventional explicit TINs and the regular grid DEM. The storage of each TIN vertex is equivalent to one two-byte regular grid elevation, if the data set is compressed or concatenated into the box-sort data structure (Figure 8.10). As such, all the tables and figures presented in this chapter which reference the storage costs of the (explicit) TIN may be rewritten or redrawn with the results for the values of '% Storage' substituted by the values for '% of Selected Grid Points' (fourth columns of Tables 8.1 and 8.2 for ST06 and ST08 respectively). The results for all applications, including profile interpolation and radio path loss prediction will be the same as for the explicit TIN. However, the computational efficiency of the Implicit TIN will decrease, since there is the additional overhead of performing triangulation 'on the fly' or at the application stage.

The triangulation algorithm developed for this research will calculate and store all the Delaunay edges for 250 data points in approximately one second of CPU time. Hence, typical profiles may take up to four seconds (for 1000 points) to construct. However, the topology of this profile sub-TIN may be reused for other profile calculations. The nature of this application (calculating multiple radio path losses) may call for other profiles to be calculated within the vicinity of the original profile. The bounds of the profile sub-TIN have to be sufficient to ensure that the original topology is retrieved. As such, in some instances profiles whose transmitter and receiver coordinates differ by up to a kilometre from the original profile, may be interpolated from the profile sub-TIN of this original profile. Furthermore, the profile sub-TIN does not have to be constrained by storage limits. This reconstructed (memory-resident) TIN can represent a greater degree of topology than the vertex-based TIN. Experimental results have shown that profile interpolation can be up to three times faster than for a vertex-based TIN with this greater degree of TIN topology.

Profile Path = 13.205 Kilometres

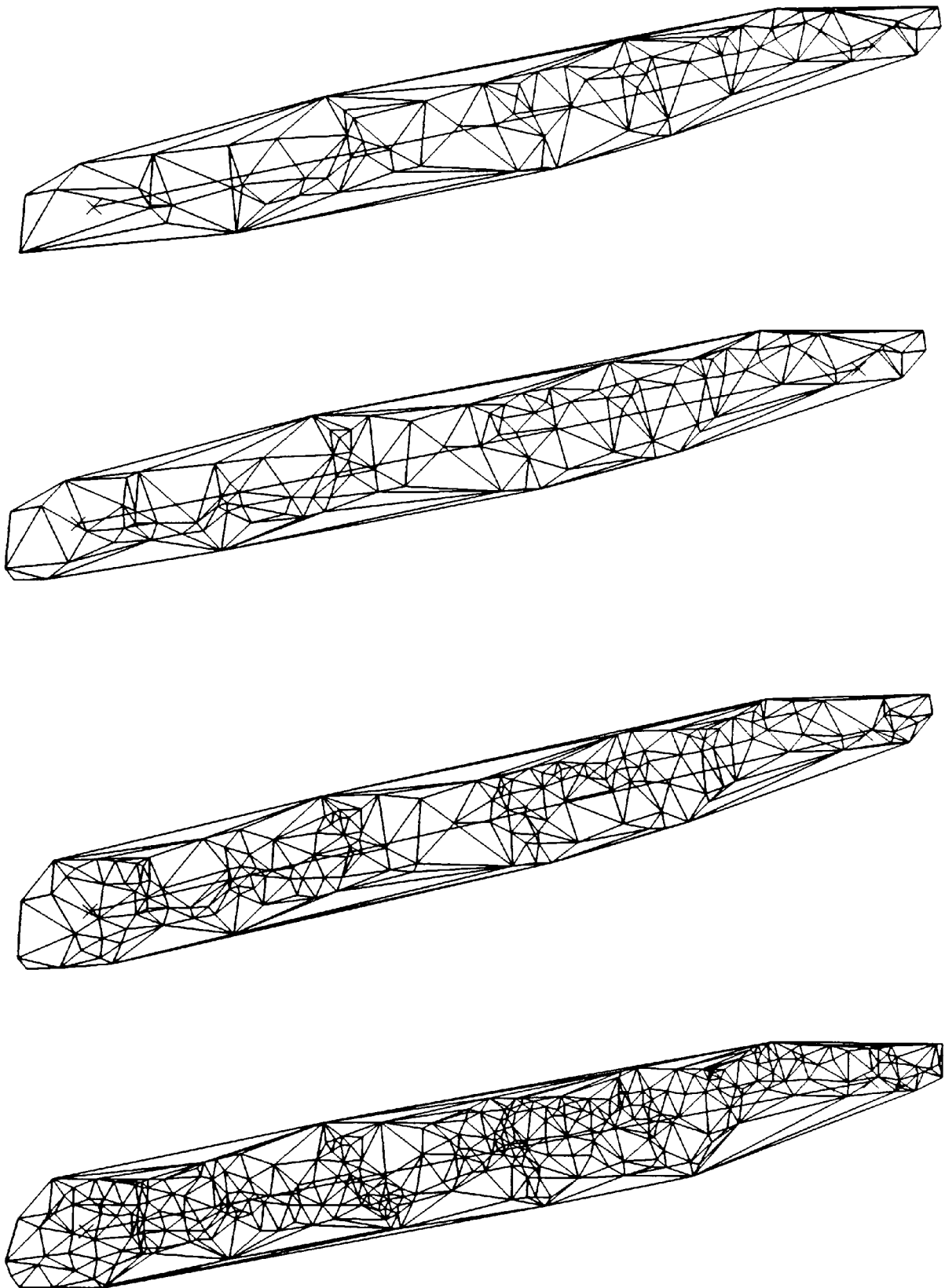


Figure 8.29 - Vertices and Implicit TINs for a Profile through ST08 at Tolerances of 25, 20, 15 and 10 Metres Maximum Absolute Error.

These Implicit TINs consist of 87, 114, 185 and 347 vertices, respectively.

The use of the Implicit TIN will however, require an efficient Delaunay triangulation algorithm. The development of such algorithms is an area that has recently received widespread attention, particularly for digital terrain modelling (Correc & Chapuis, 1987; Collins, 1989; and ElGindy, 1990). For very large data sets, fast computation of the Delaunay triangulations is essential. Whilst some algorithms (Correc & Chapuis, 1987; Dwyer, 1987) suggest significant improvements on earlier algorithms, triangulations are most efficiently constructed using a parallel algorithm (ElGindy, 1986, 1990; Merks, 1986; Collins, 1989; Ware & Kidner, 1991). Furthermore, the uniqueness of the Delaunay triangulation facilitates such a parallel approach.

Chapter Nine

Summary of Results and Conclusions

9.1 Introduction

This thesis illustrates the applicability and suitability of various methods of storing terrain data. In particular, their efficiency for retrieving profiles for estimating radio path propagation in a mobile communication system has been examined. Whilst these digital elevation models are only a selection of the many methods available, each has been chosen as a representative of a distinct and popular means of surface modelling. Some of these models have been constructed in a novel form, specifically for the purposes of this research. In addition, two new DEMs have been described and implemented which enable extensive storage savings to be attained over other DEMs and the regular grid in particular. The Implicit TIN (Chapter Eight and Kidner & Jones, 1991) and the Huffman-encoded grid (Chapter Five and Kidner & Smith, 1991) are both variations on existing, popular DEMs (the TIN and the regular grid). Both of these models extend the attractiveness of these methodologies by offering significant reductions in storage requirements. This characteristic is not only of great benefit in a mobile system, where 'ruggedised' storage costs are particularly expensive, but for all terrain modellers who require access to a large number of data sets.

For the purpose of this study, the important characteristics of a DEM have been categorised by the fidelity of the terrain representation (elevation error), storage efficiency, computational efficiency of path profiling and application suitability with respect to the features of the radio path propagation algorithm. This last criterion also highlights some important characteristics which may not necessarily be identified by an examination of elevation error only. For example, the radio path loss prediction algorithm gives a good overall impression of terrain model performance in a similar manner to visualisation applications, and in particular examines the integrity of critical features such as ridges and peaks.

At this stage, the reader may not be aware of the DEM which best fits these overall criteria. Each of the DEMs in the preceding chapters offers significant advantages in some respect, but also exhibit some disadvantages. In general, these disadvantages may be the elevation performance; the computational efficiency for profile retrieval; or insignificant storage savings. For the purposes of this research, a 'significant' storage saving has been categorised as being 'in excess of 75%' of the original grid DEM storage. This storage convention was adopted in order to identify a viable alternative DEM. Unless this criterion is achieved, there is little likelihood of a DEM becoming universally accepted for the required application in this study. Hence, a small storage saving will offer no significant advantage over the commonly used regular grid DEM.

In most instances, digital terrain modellers are faced with the dilemma of choosing between a DEM whose primary characteristic is storage efficiency or computational efficiency for the required application. This assumes that storage efficient models are not computationally efficient. This is not necessarily true in all instances (for example, mathematical models such as a polynomial-based DEM), but generally, storage efficient models remove topological relationships, thus hindering performance, such as search speed. This decrease in model performance or computational efficiency cannot be effectively measured before detailed analysis of prototype terrain models has been accomplished. For this reason, alternative DEMs were primarily considered for this study on the basis of offering significant storage savings over the regular grid DEM. Once selected, prototype models were implemented to determine their computational efficiency for profiling.

The sampling of a regular grid DEM is a key issue in digital terrain modelling, but has not been given the attention that it deserves. National mapping agencies such as the Ordnance Survey (O.S.) produce regular grid data sets aligned to the National Grid coordinates at a uniform sampling interval of 50 metres, irrespective of terrain variability. Hence, the major source of elevation data for the United Kingdom is not adaptable to terrain variability. As a result, the accuracy to which some applications perform may be severely affected by an under-sampling of points in variable terrain. Furthermore, the storage costs of such regular grid DEMs may be prohibitively high, due to an over-sampling of points or data redundancy. This factor is significantly exaggerated by the large number of coastal data sets for the U.K., many of which have 50% or more sea level values.

For many applications for which the O.S. data is required, the density of grid sampling will not be a major problem. For the intended application of this research, both of the problems highlighted above are significant, namely storage and accuracy of the predicted radio path loss. Firstly, data redundancy cannot be tolerated and must be overcome with a storage efficient model which is adaptive to terrain variability. Whilst the choice of a sparser grid will be more storage efficient, it will produce unconstrained elevation errors. Secondly, whilst some researchers advocate the use of sparse grids for radio path prediction (for example, the 500 metre grid of Edwards & Durkin, 1969), the results shown here prove that radio path loss prediction is critically affected by changes in the density of grid sampling. This is largely dependent upon the sensitivity of the algorithm used. For ST08, a 250 metre grid produces up to 25% of random profiles with an absolute radio path loss error greater than 6 dBs, when compared to 0% for a 50 metre grid (see Table 9.2). With a 100 metre grid, the number of such profiles is 6% to 7%. These errors in relation to the grid sampling density suggest that up to 7% of original (50 metre grid) profiles may produce such radio path loss errors if a smaller sampling density (ie. 25 metre grid) were adopted. This highlights the problem of being constrained to a uniform regular grid DEM. In any such DEM there will always be errors

between the digital representation and the 'real world'.

In particular, radio path loss prediction in a 50 metre grid will produce a number of errors due to a variety of factors. Firstly, the accuracy of the data acquisition and grid interpolation can produce significant elevation errors in these source DEMs (Morris & Flavin, 1990). Associated with the data acquisition problem is the lack of detailed information on vegetation such as trees and buildings which may cause clutter loss. Also, the effect of grid sampling interval has not been fully investigated or compared with 'real world' results. The results presented here suggest that this factor could be just as critical. Each of these problems is costly to remedy, in terms of both expense and labour, and hence impractical for the nature of this study. More significantly, the difficulty of taking accurate propagation measurements in the field hinders the choice of (1) the most accurate grid DEM which is best suited for the radio path loss algorithm, irrespective of storage costs, and (2) the best alternative DEM which provides storage efficiency. The answer to the first question must surely be the grid which provides the most detail or information. This is the the grid with the densest sampling interval, which for the foreseeable future will be the 50 metre grid DEM (in the U.K.), despite its considerable shortcomings. As such, for the purposes of this study, this DEM was adopted as the 'real world' model against which the search for an answer to (2) is based.

This chapter will briefly review the overall performances of the implemented DEMs, so that distinct comparisons can be made. The features of the DEMs that are critical for efficient performance are summarised and suggestions are made as to how future improvements may be made. Conclusions are then drawn together to determine the digital elevation model which 'best' fits the application of calculating radio path propagation in a mobile communications network.

9.2 Summary of Results

At this stage it is necessary to summarise the performance of each of the digital elevation models in a comparable format. This is accomplished with the aid of abridged tables representing the major performance features for the data sets ST06 and ST08 (Tables 9.1 & 9.2). These tables are in no way complete, but represent a selection of 39 examples of the digital elevation models described in the earlier chapters. For easier comparison, the performance results of the radio path propagation have been condensed. At each of the radio frequencies over which the models were tested ($f=200, 400, 600, 900, 1400$ and 1800 MHz), the results have been averaged to formulate one statistical measure, rather than six for each model.

One feature of Tables 9.1 and 9.2 which hinders a distinct comparison between DEMs is the time taken for profile interpolation. The results represent the average interpolation time for a

typical profile of 10 to 12 kilometres on a DEC VAX 8650. However, in many cases there is an additional overhead for a 'one-off' retrieval of the regular grid DEM. Profile interpolation within such a grid is computationally efficient, since the grid coordinates are implicit within the data structure, thus providing an efficient search mechanism. Many of the alternative DEMs presented within this study cannot perform to a similar level of efficiency. The most storage efficient DEMs achieve their savings by sacrificing ease of retrieval or analytical flexibility. This is usually accomplished by removing much of the topology which is necessary for computationally efficient applications. Therefore, for some DEMs, multiple profile interpolation is better served by transforming the original data structure into a more flexible form, such as a regular grid. The time taken for generating this temporary data structure will in most instances be more tolerable than inefficient profile interpolation.

For example, consider the Huffman-encoded regular grid DEM. Profile interpolation within this data structure is too dependent upon the path of the profile. In the worst case, the complete data set may have to be decoded. Therefore, it is more feasible to retrieve the compressed data into a memory-resident grid DEM, for multiple profile interpolation. This is true for all data-compressed DEMs based upon the regular grid data structure.

For each of the distinct classes of DEMs presented in Tables 9.1 and 9.2, a summary of profile interpolation is presented below. This should be read in conjunction with the results presented in Tables 9.1 and 9.2, to clarify the comparison of profile times.

- (a) The **regular grid DEM** is the most efficient data structure for many applications, such as interpolation. In a 50 metre grid, a typical uniformly sampled profile of 10 to 12 kilometres (200 to 240 points) can be interpolated in under 0.3 milliseconds. The time taken for sparser grids is directly related to the sampling interval, or more specifically the number of interpolations.
- (b) Interpolation in a **variable density grid** can be performed at uniform intervals (ie. 50 metres) or at the grid cell boundaries. However, the former method is simpler. Even though there may be redundant points in the profile, interpolation time will be the same as for the regular grid DEM.
- (c) Interpolation in the **differential altitude grid** can be performed directly from the compressed grid cells in 13.5 milliseconds (45 times slower than the regular grid), or by decoding the complete regular grid in approximately one second and interpolating multiple profiles. Thus, if more than 80 profiles are required a time saving could be achieved by decoding the differential grid into a regular grid DEM.
- (d) Direct profile interpolation is impractical within any **Huffman-encoded regular or**

variable density grid. The data structure should always be decoded into a conventional regular grid DEM. The time overhead for performing this task is related to the average code length and the number of vertices, or more specifically, the terrain variability. This maximum time for an error free data set of variable terrain is 5 seconds for 160,000 vertices.

- (e) Uniform profile interpolation within a **polynomial surface patch DEM** is very efficient, due to its simplicity. The degree of the polynomial has only a small effect on interpolation time. In general, typical profiles for polynomials of degree two to six can be interpolated in under 0.35 milliseconds.
- (f) **Adaptive polynomial surface patches** of up to degree 10 are just as efficient as fixed-degree polynomials or the regular grid DEM, provided there is an index to the polynomial degree for each patch. The interpolation time for 10th degree polynomials is not too significant and is compensated for by the large number of polynomials of low order. Average interpolation time is under 0.35 milliseconds.
- (g) The **surface patch quadtree** data structure is inefficient for profile interpolation, due to the compressed indexed key for the nodes (which is needed for storage efficiency) and the complex algorithm needed for following the path of the profile. A number of nodes which are not intersected will therefore need to be decoded. For multiple profile interpolation, it is therefore more practical to reconstruct a regular grid DEM by interpolating the 256x256 vertices from the surface functions of each quadtree node. This can be accomplished in a maximum of 1.7 seconds of CPU time at the 5 metre tolerance for data set ST08.
- (h) The **vertex-based TIN** requires a significant degree of search time to follow the path of the profile through the network of triangle edges. Average interpolation time is 12 milliseconds for the densest TIN, or 40 times greater than the regular grid DEM. However, there are no redundant profile points, since all points are interpolated at critical changes in slope.
- (i) The **Implicit TIN** is not a computationally efficient DEM, since the topology needs to be reconstructed for each profile path. This time will depend upon the variability of the terrain within this region. The algorithm takes approximately one second to triangulate 250 data points. For the densest TIN, a profile sub-TIN which includes a suitable boundary for multiple paths may consist of 1000 vertices (ie. 4 seconds to construct). However, since storage of this temporary sub-TIN is not as important, a triangle-based data structure can be used. Profile interpolation can be up to three times faster than for the vertex-based TIN.

Digital Elevation Model			% Storage	RMSE (m)	% Points within ± 10 m	Profile Time	RMSE (dB)	% Profiles within ± 6 dB
Regular Grid	(1)	50 m	100.0	0.00	100.0	0.28 ms	0.00	100.0
	(2)	100 m	25.13	0.99	99.91	0.14 ms	2.29	95.9
	(3)	150 m	11.22	1.61	99.66	0.09 ms	3.30	92.3
	(4)	200 m	6.34	2.22	99.14	0.07 ms	3.99	89.2
	(5)	250 m	4.08	2.88	98.29	0.06 ms	4.76	83.5
Variable Density Grids	(6)	5 m	27.15	0.99	100.0	0.28 ms	1.73	97.8
	(7)	10 m	12.44	2.13	100.0	0.28 ms	3.03	93.0
	(8)	15 m	8.20	2.92	98.76	0.28 ms	3.80	89.8
Differential Altitude Grids	(9)	2-bit	16.00	1.43	99.70	13.5 ms or	2.14	96.7
	(10)	3-bit	24.00	0.77	99.99	1.0 s + 0.28 ms	1.70	97.6
Huffman Encoded Grids	(11)	0.0 m	12.30	0.00	100.0	3.0 s + 0.28 ms	0.00	100.0
	(12)	0.5 m	9.81	0.40	100.0	2.5 s + 0.28 ms	1.28	98.8
	(13)	1.5 m	8.48	0.90	100.0	2.0 s + 0.28 ms	1.92	97.0
	(14)	2.5 m	7.98	1.37	100.0	1.5 s + 0.28 ms	2.45	95.5
Sub-Sampled Huffman Grids	(15)	100 m	4.37	0.99	99.91	1.1 s + 0.28 ms	2.29	95.9
	(16)	200 m	1.49	2.22	99.14	0.5 s + 0.28 ms	3.99	89.2
Variable Density Huffman Grids	(17)	2.5 m	12.22	0.42	100.0	2.0 s + 0.28 ms	1.04	99.2
	(18)	5.0 m	7.47	0.99	100.0	1.5 s + 0.28 ms	1.73	97.8
	(19)	7.5 m	5.17	1.59	100.0	1.3 s + 0.28 ms	2.44	95.5
	(20)	10.0 m	3.90	2.13	100.0	1.0 s + 0.28 ms	3.03	93.0
Polynomial Surface Patches	(21)	4th (4d)	7.50	2.67	98.80	0.30 ms	3.69	91.0
	(22)	5th (5d)	10.50	2.41	99.12	0.34 ms	3.66	91.5
	(23)	5th (5c)	16.41	1.76	99.66	0.34 ms	2.73	94.5
	(24)	3rd (3c)	20.00	1.61	99.74	0.28 ms	2.81	94.9
Adaptive Polynomials	(25)	10 m	7.71	2.13	99.99	0.32 ms	2.90	93.7
	(26)	15 m	4.66	3.18	98.44	0.31 ms	3.58	91.3
	(27)	20 m	3.40	4.09	95.84	0.30 ms	3.56	91.0
Surface Patch Quadtrees	(28)	5 m	14.59	1.09	100.0	0.7 s + 0.28 ms	2.01	96.7
	(29)	10 m	6.75	2.33	100.0	0.4 s + 0.28 ms	3.13	92.1
	(30)	15 m	3.69	3.59	97.67	0.3 s + 0.28 ms	4.27	85.8
	(31)	20 m	2.23	4.99	92.28	0.2 s + 0.28 ms	5.48	79.3
TIN	(32)	5 m	28.61	1.27	100.0	4.2 ms	2.11	96.8
	(33)	10 m	14.93	2.59	100.0	2.1 ms	3.27	91.8
	(34)	15 m	9.01	3.93	97.04	1.3 ms	4.35	85.3
	(35)	20 m	6.52	5.71	90.37	0.9 ms	5.23	79.9
Implicit TIN	(36)	5 m	2.89	1.27	100.0	2.2 s + 1.4 ms	2.11	96.8
	(37)	10 m	1.52	2.59	100.0	1.0 s + 0.9 ms	3.27	91.8
	(38)	15 m	0.93	3.93	97.04	0.6 s + 0.6 ms	4.35	85.3
	(39)	20 m	0.68	5.71	90.37	0.4 s + 0.4 ms	5.23	79.9

Table 9.1 - Abridged Summary of Results for 39 Digital Elevation Models of ST06.

(Average Profile Time is measured in Milliseconds. The time in seconds corresponds to the reconstruction time for a more computationally efficient temporary data structure).

Digital Elevation Model			% Storage	RMSE (m)	% Points within ± 10 m	Profile Time	RMSE (dB)	% Profiles within ± 6 dB
Regular Grid	(1)	50 m	100.0	0.00	100.0	0.28 ms	0.00	100.0
	(2)	100 m	25.13	1.82	99.69	0.14 ms	3.41	93.3
	(3)	150 m	11.22	3.11	98.29	0.09 ms	4.70	87.6
	(4)	200 m	6.34	4.48	95.14	0.07 ms	5.68	80.5
	(5)	250 m	4.08	6.13	90.28	0.06 ms	6.84	75.2
Variable Density Grids	(6)	5 m	72.16	1.03	100.0	0.28 ms	1.57	98.2
	(7)	10 m	30.57	2.73	100.0	0.28 ms	3.01	93.4
	(8)	15 m	15.93	4.21	96.73	0.28 ms	4.43	87.7
Differential Altitude Grids	(9)	2-bit	16.00	3.34	97.86	13.5 ms or	3.13	94.2
	(10)	3-bit	24.00	2.28	99.83	1.0 s + 0.28 ms	2.84	95.5
Huffman Encoded Grids	(11)	0.0 m	20.24	0.00	100.0	5.0 s + 0.28 ms	0.00	100.0
	(12)	0.5 m	15.31	0.50	100.0	3.6 s + 0.28 ms	1.73	98.2
	(13)	1.5 m	11.49	1.12	100.0	3.0 s + 0.28 ms	1.87	97.7
	(14)	2.5 m	10.34	1.71	100.0	2.7 s + 0.28 ms	2.58	95.7
Sub-Sampled Huffman Grids	(15)	100 m	7.00	1.82	99.69	1.8 s + 0.28 ms	3.41	93.3
	(16)	200 m	2.29	4.48	95.14	0.7 s + 0.28 ms	5.68	80.5
Variable Density Huffman Grids	(17)	2.5 m	32.52	0.33	100.0	3.7 s + 0.28 ms	0.74	99.7
	(18)	5.0 m	23.50	1.03	100.0	2.5 s + 0.28 ms	1.57	98.2
	(19)	7.5 m	16.16	1.88	100.0	2.0 s + 0.28 ms	2.36	95.6
	(20)	10.0 m	11.40	2.73	100.0	1.5 s + 0.28 ms	3.01	93.4
Polynomial Surface Patches	(21)	4th (4d)	7.50	5.73	91.74	0.30 ms	5.81	80.4
	(22)	5th (5d)	10.50	5.10	94.00	0.34 ms	5.18	82.9
	(23)	5th (5c)	16.41	3.33	98.36	0.34 ms	3.99	89.6
	(24)	3rd (3c)	20.00	3.03	98.78	0.28 ms	3.56	90.8
Adaptive Polynomials	(25)	10 m	20.24	2.64	99.96	0.36 ms	3.18	93.1
	(26)	15 m	11.73	3.89	98.10	0.33 ms	4.18	88.5
	(27)	20 m	8.06	5.13	93.95	0.31 ms	4.79	85.3
Surface Patch Quadtrees	(28)	5 m	35.88	1.22	100.0	1.7 s + 0.28 ms	2.06	97.2
	(29)	10 m	17.94	2.71	100.0	0.9 s + 0.28 ms	3.36	92.7
	(30)	15 m	11.37	4.12	96.88	0.6 s + 0.28 ms	4.27	88.3
	(31)	20 m	8.01	5.59	90.71	0.5 s + 0.28 ms	5.25	82.9
TIN	(32)	5 m	83.24	1.49	100.0	12.0 ms	2.11	97.1
	(33)	10 m	42.07	3.27	100.0	5.6 ms	3.45	91.1
	(34)	15 m	24.88	4.78	95.63	3.6 ms	4.70	86.8
	(35)	20 m	17.77	6.22	88.29	2.5 ms	5.50	81.5
Implicit TIN	(36)	5 m	8.37	1.49	100.0	4.0 s + 4.0 ms	2.11	97.1
	(37)	10 m	4.25	3.27	100.0	2.7 s + 2.0 ms	3.45	91.1
	(38)	15 m	2.53	4.78	95.63	1.5 s + 1.6 ms	4.70	86.8
	(39)	20 m	1.82	6.22	88.29	1.0 s + 1.1 ms	5.50	81.5

Table 9.2 - Abridged Summary of Results for 39 Digital Elevation Models of ST08.

(Average Profile Time is measured in Milliseconds. The time in seconds corresponds to the reconstruction time for a more computationally efficient temporary data structure).

Each of the entries in the tables is measured in comparison to the 50 metre regular grid DEM (entry 1 in each table). For example, Model 2 in Table 9.1 represents a 100 metre regular grid, which requires only 25.13% storage of the original grid, whilst profiles are constructed in half of the original time, with 95.9% of these being acceptable for predicting the radio path loss to within an absolute error of 6 dBs.

Both of these tables are consistent with one another, so that the performances of nearly all 39 DEMs are comparable. The models in which some performance criteria are not as consistent as one would expect are highlighted later in this chapter. Since ST06 (Table 9.1) and ST08 (Table 9.2) represent two contrasting data sets, it is expected that the performance of models for other terrain would lie somewhere between these two extremes. The majority of British terrain (and England in particular) is not as variable as ST08, whilst few data sets would be as 'data redundant' as ST06.

When comparing the performances of these terrain models, any accepted alternative to the regular grid DEM should offer significant improvements, especially in terms of storage. In this respect, the regular grid should be considered at any sampling interval and not constrained to just the original 50 metre point density. Any alternative terrain model should not only offer significant improvements over the 50 metre grid, but should have significant advantages over the 100 metre grid as well. Hence, the performance of the regular grid DEM and the sub-sampled grids derived from this should therefore determine the standard against which other models are compared. This, together with Tables 9.1 and 9.2 can be more clearly illustrated with the aid of performance graphs. From these graphs, it is possible to determine the terrain models which offer the greatest benefits. Other models which perform comparably with the regular grid at various resolutions may be rejected, since such a sub-sampled grid could be more conveniently adopted.

The following graphs (Figures 9.1 to 9.3) illustrate the performance of the DEMs for ST08, in terms of R.M.S. elevation error, R.M.S. path loss error and percentage of profiles within ± 6 dB. The results for the latter two graphs are averaged over 1000 random profiles over frequencies in the range of 200 to 1800 MHz. The graphs include the data for all of the DEMs considered in the earlier chapters and not just the results for the 39 DEMs presented in Table 9.2. It should be noted that the graphs for ST06 behave in a similar fashion to those of ST08. To illustrate the performance of the methods clearly (and to reduce clustering of values), separate graphs are drawn for the DEMs that use surface functions (mathematical models) and those which directly store the elevations (point models). The X-axis of each graph represents the storage requirements of the terrain models as a percentage of the original two-byte regular grid DEM. In each of these graphs, the results for the regular grid DEM (and sub-sampled grid) are displayed as the standard to which the other models are compared.

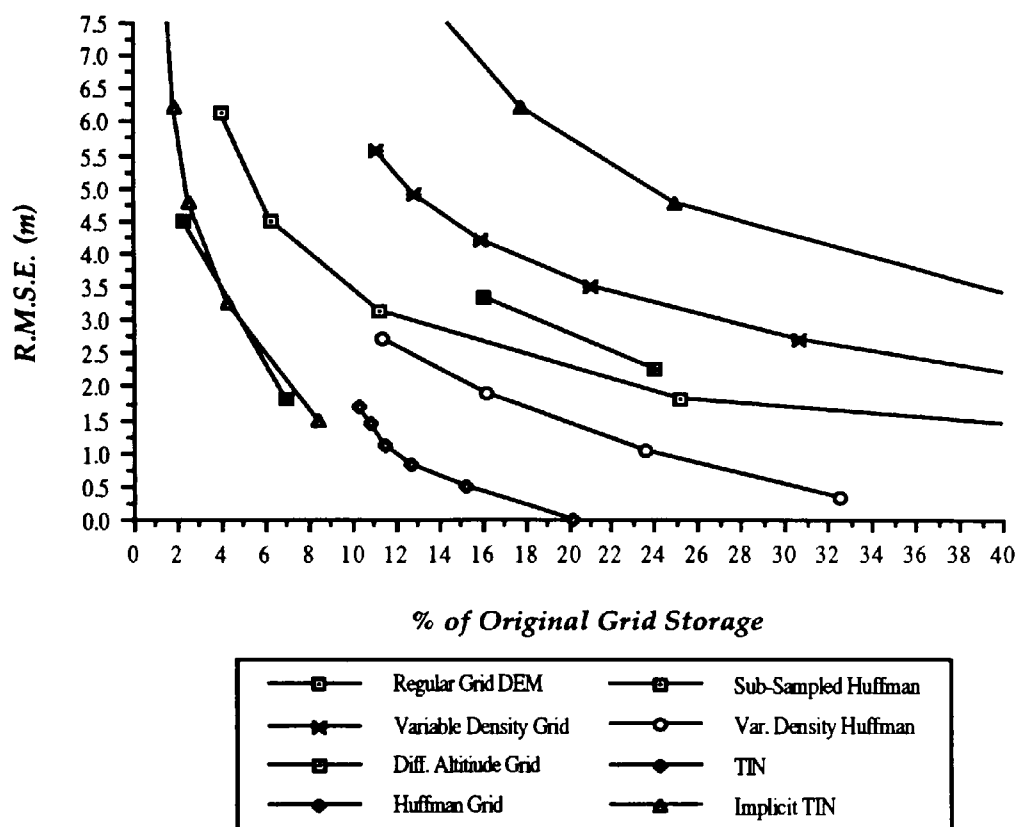


Figure 9.1a - ST08 Elevation RMSE v. Storage (Point DEMs)

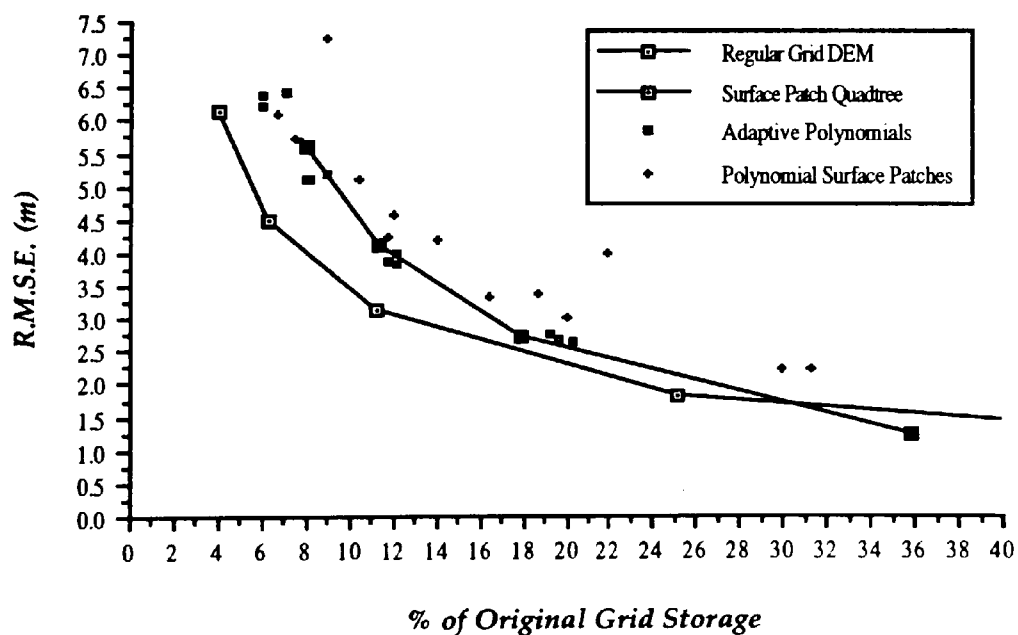


Figure 9.1b - ST08 Elevation RMSE v. Storage (Mathematical DEMs).

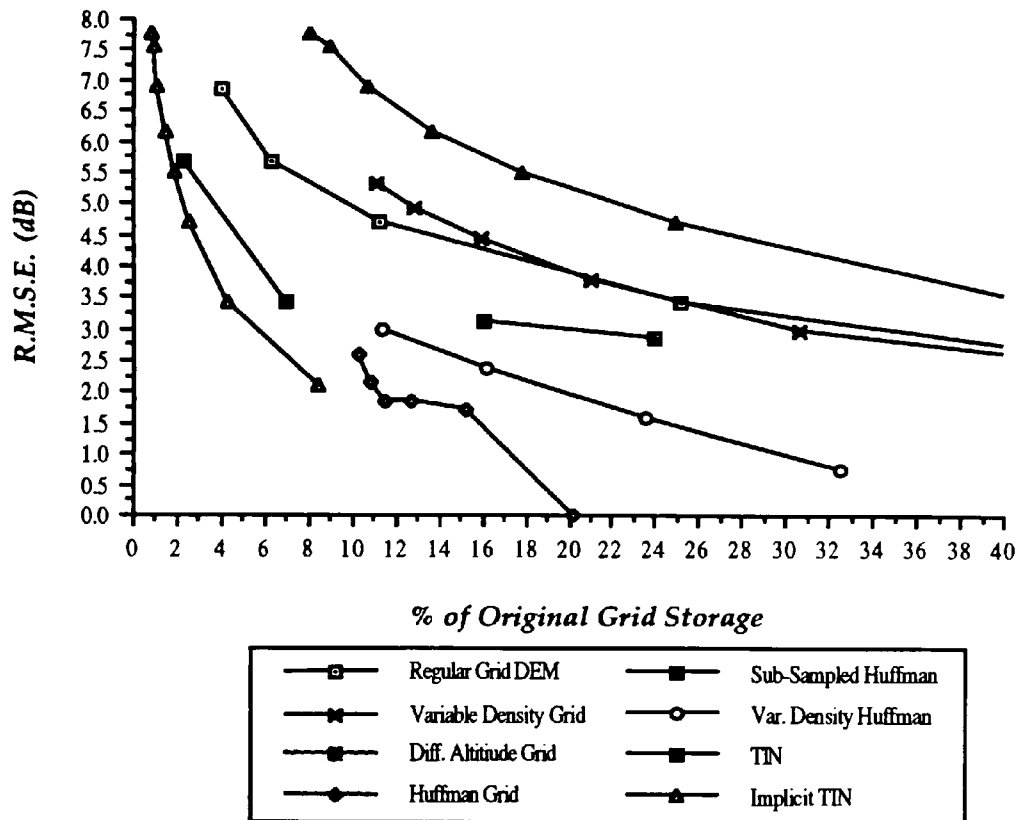


Figure 9.2a - ST08 Radio Path Loss RMSE v. Storage (Point DEMs)

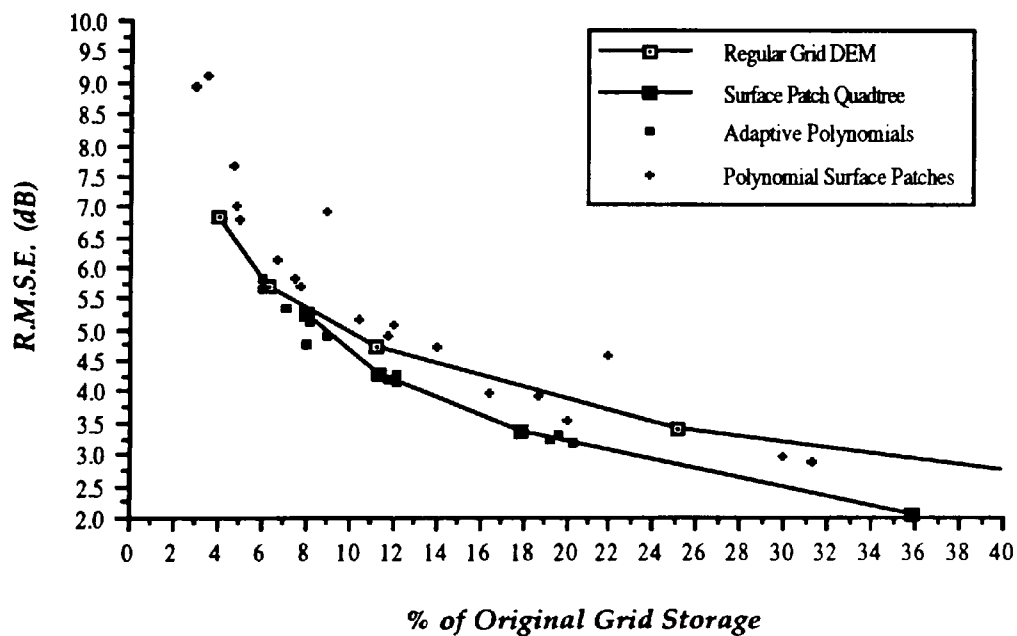


Figure 9.2b - ST08 Radio Path Loss RMSE v. Storage (Mathematical DEMs)

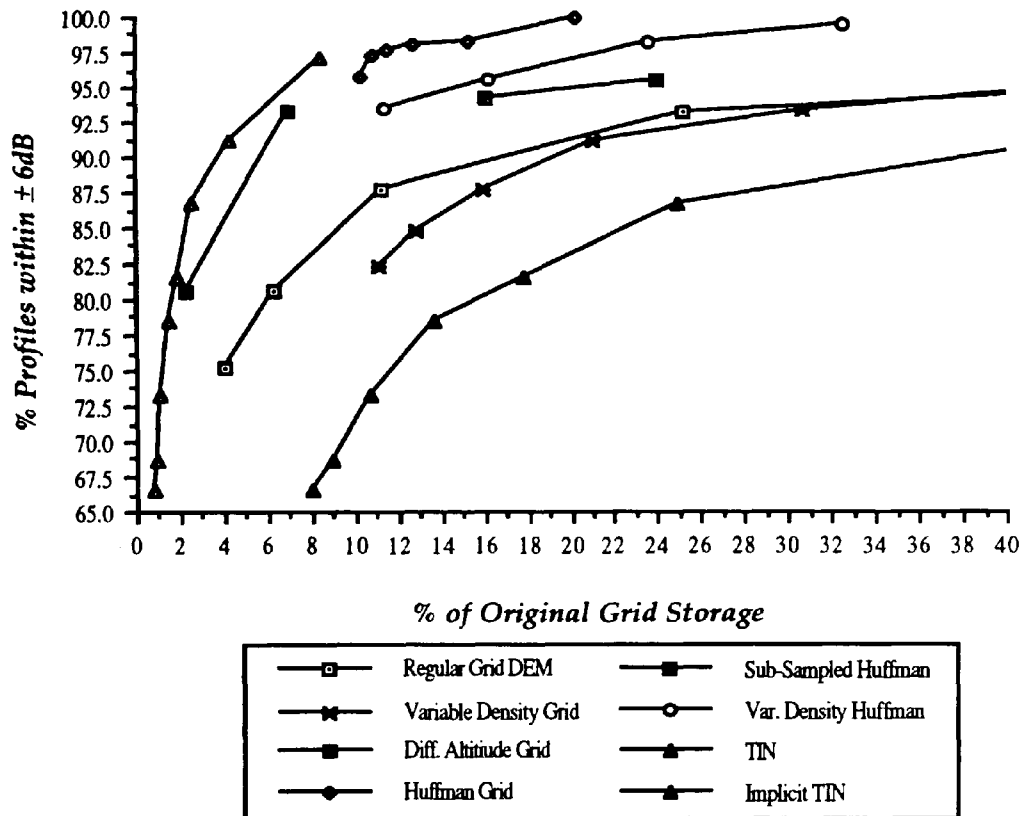


Figure 9.3a - Percentage of ST08 Profiles Within a Radio Path Loss Error of ± 6 dB (Point DEMs)

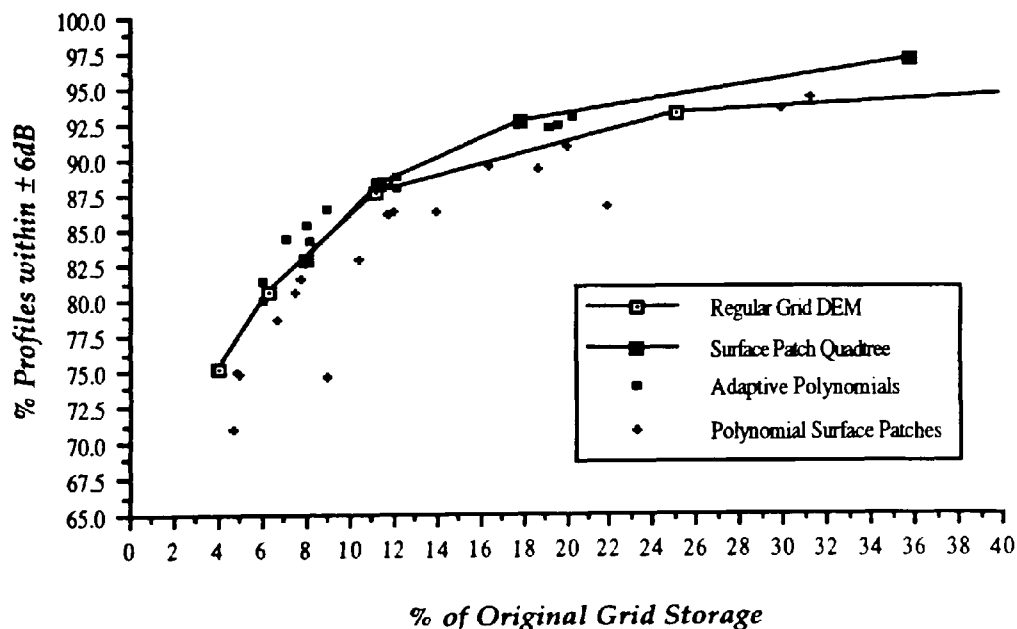


Figure 9.3b - Percentage of ST08 Profiles Within a Radio Path Loss Error of ± 6 dB (Mathematical DEMs)

In terms of R.M.S. elevation error against storage (Figures 9.1a and 9.1b), there are only four models which give a better performance than the regular grid at various resolutions. These are shown in Figure 9.1a as the values beneath the graph of the sub-sampled regular grids. All of these models are data compressed DEMs - the Huffman encoded regular, variable and sub-sampled grids and the Implicit TIN. All other models offer no significant advantages over the regular grid.

In terms of R.M.S. radio path loss errors (Figures 9.2a and 9.2b) and percentage of satisfactory profiles (Figures 9.3a and 9.3b) however, not only do the compressed DEMs outperform the regular grids, but the differential altitude grids, surface patch quadtrees, adaptive polynomials and some regular polynomial surfaces give a better radio path loss performance. It is noticeable therefore, that some terrain models possess attributes which are better suited for the calculation of radio path loss than applications which are solely based on elevation, such as contouring. For example, in some instances, very large elevation errors in areas of steep slopes, such as cliffs, may have very little or no effect on certain profiles in terms of radio path loss error. Hence, elevation error is not necessarily the best estimate for the evaluation of DEM performance.

Whilst many of the DEMs give a good overall radio path loss prediction performance, the results need to be examined in a wider context. It should be remembered that the simplest alternative to the dense regular grid is a sub-sampled grid. Hence, the DEMs in Figures 9.2 and 9.3 which improve upon the performance of the regular and sub-sampled grids, should do so by a significant or worthwhile margin. To this extent, the mathematical models (polynomials and surface patch quadtree) can be excluded.

As already seen, the graphs in Figures 9.1 to 9.3 play an important role in the choice of an alternative DEM for radio path loss estimation. However, one important factor which will expedite this decision has been omitted from these graphs. This is the computational efficiency of the DEMs, or more specifically for this application, the time taken to generate the elevation profiles. Ideally, the choice of DEM is one which is both storage efficient and computationally efficient for determining radio path losses with relatively few errors. Ideal parameters determining efficiency could be 95% of profiles within a path loss error of ± 6 dBs; a 75% storage saving over the regular grid and an average profile generation time of less than 1 millisecond. Regular grid profile generation time is approximately one third of this, for a typical 12 kilometre path sampled every 50 metres.

Of all the DEMs researched and described in this study, no model fits these ideal criteria for both ST06 and ST08. The one area in which promising models consistently fail to meet these standards is the time taken for profile generation. A summary of the DEMs and their

performance (for ST08) is presented below in Table 9.3. This table also includes the criterion that the DEM should be constrained by maximum absolute error.

DEM	Max. Error Constrained	100% Pts. < $\pm 10\text{m}$ & Storage < 25%	95% Profs. < $\pm 6\text{dB}$ & Storage < 25%	Time Efficient
Sub-Sampled Grids	X	X -	X -	✓
Variable Density Grids	✓	X -	X -	✓
Differential Alt. Grids	X	X -	✓ 24.00%	X
Huffman-Encoded Grid	✓	✓ 10.34%	✓ 10.34%	X
Sub-Sampled Huffman	X	X -	X -	X
Variable Density Huffman	✓	✓ 11.40%	✓ 16.16%	X
Polynomials	X	X -	X -	✓
Adaptive Polynomials	✓	✓ 20.24%	X -	✓
Surface Patch Quadtree	✓	✓ 17.94%	X -	X
TIN	✓	X -	X -	X
Implicit TIN	✓	✓ 4.25%	✓ 8.37%	X

Table 9.3 - Summary of Overall Results for ST08 and Attainment of Criteria for an Ideal DEM.

It is apparent therefore, that since no single model fits all of these ideal requirements, then the subjective criteria for the selection of an alternative DEM need to be reappraised. Storage must always be one of the primary concerns regarding this decision. However, many of the DEMs for ST08 perform consistently better than the guideline of a 75% storage saving and are also constrained to maximum elevation tolerances (Table 9.3). Therefore, the criterion that must be relaxed in order to determine the ideal DEM is either radio path loss performance or computational efficiency.

In the first instance, consider radio path loss error. If the assumption that 95% of profiles must be within an error of ± 6 dBs is revised to a less constrained total of 90% (and the DEM is storage and computationally efficient), the ideal DEM is a polynomial model of either uniform or variable degree (models 24 and 25 in Table 9.2), or the 100 metre sub-sampled grid. Despite the additional storage savings offered by the polynomial models, for the benefit of simplicity, the sub-sampled grid would appear to best-fit the requirements of the ideal model. However, it has already been shown that a sub-sampled grid does not constrain the maximum elevation error; the model is not adaptable to terrain variability; and the effects of grid cell size on radio path loss have not been fully investigated with 'real world' results. As such, radio path loss error may vary substantially for different types of terrain. It is natural to assume therefore, that for some data sets a 100 metre grid DEM will not produce 90% of profiles within an error of ± 6 dBs. Hence, an unconstrained model, such as a sub-sampled grid or a DEM of uniform

polynomial surface patches, will always have an associated degree of unpredictability. This may also be a problem for the adaptive polynomial DEM, since even a 10th degree polynomial sometimes cannot accurately fit some surface patches.

Whilst there are DEMs which are both storage and computationally efficient, these tend to be unconstrained in elevation error, with a resulting unpredictability in radio path loss performance. As such, some models may produce only 80% to 90% of profiles within an acceptable path loss error of ± 6 dBs. The alternative is to examine DEMs which are both storage efficient and produce consistent radio path loss estimates. In doing so, the criterion of computational efficiency must be relaxed. This can perhaps be considered the least important criterion, since it does not directly affect the elevation or path loss results, such that it has no direct influence in the choice of siting a radio transmitter or receiver.

Bilinear interpolation is an extremely efficient operation which contributes significantly to the popularity of the regular grid as a DEM. Throughout this study, it has been used as the benchmark against which computational efficiency is examined. Since grid coordinates are implicit, the efficiency of operations which include searching can only be matched by other grid-based DEMs. Whilst the compressed DEMs such as the differential altitude and Huffman grids are based upon the regular grid, computational efficiency is severely hindered by the reconstruction time needed to retrieve the grid into computer memory. As such, this 'one-off' retrieval overhead makes compressed DEMs unattractive for profile retrieval. However, in estimating radio path loss for the siting of a radio network, many possible profiles will need to be examined as suitable paths and interference paths. In such circumstances, there is a likelihood that a one-off overhead may be tolerated. The results for ST08 suggest that this would be a maximum of five seconds of CPU time for every DEM, in the worst case (ie. an error-free Huffman grid). All profiling operations within such a grid would then be executed in an average time of under 0.3 milliseconds. This same argument also holds for the the implicit TIN, since a local triangulation is performed along the path of the profile. Since profile calculations are likely to be in the same vicinity, the explicit TIN framework generated by the first profile should be suitable for other subsequent profiles.

An additional consideration that is worth further attention is the computational efficiency of the radio path loss algorithm itself. A typical profile will require approximately 4 milliseconds of CPU time to determine the associated radio path loss, which is nearly 15 times greater than regular grid profile interpolation time. This substantially reduces the relative time difference between regular grid DEMs and data compressed DEMs for profiling. As such, the importance placed on profile retrieval time is not as significant as first stated (ie. as efficient as the regular grid DEM). In all the implemented models, average profile retrieval time and radio path loss estimation will be measured in milliseconds rather than seconds. Only

in the most exceptional circumstances will time be such an important criterion that a maximum overhead of 5 CPU seconds cannot be tolerated.

9.3 DEM Summary

Regular Grid DEM

The regular grid continues to be the most popular form of DEM, due to a combination of attributes, including simplicity, computational efficiency and availability of data sets in this format. However, the disadvantages of the regular grid are also significant. It may create extensive data redundancy due to its inability to adapt to the terrain variability; be unable to represent critical terrain features; and be inaccurate depending upon the method of data acquisition. This last aspect is further constrained by the choice of grid size. The increasing availability of such data by national mapping agencies illustrates the need for more storage efficient DEMs. The future of such a model may therefore be dependent upon efficient data compression, such as Huffman coding.

Sub-Sampled Regular Grids

Grid sub-sampling is a regressive step in overcoming the storage deficiencies of the dense regular grid DEM. The arbitrary nature of point elimination does not take into account the variability of the terrain, thus generating a random degree of error within the model. The effect of this is largely dependent upon the original grid cell size and the nature of the terrain. A weighted or generalised reduction of grid vertices does not improve overall error performance. As such, it is not recommended as a viable alternative, even though it may appear to be the simplest solution.

Sub-sampling illustrates the excessive data redundancy within dense regular grid DEMs. For example, the 100 metre sub-sampled DEM results in less than 7% of profiles with a path loss error of more than ± 6 dBs. These results illustrate the need for reducing redundant data which do not contribute much to the overall surface representation. Since 75% of vertices can be eliminated without creating 'excessive' errors, this suggests that sub-sampling a regular grid is viable, provided it is constrained to terrain variability. The results for ST06 and ST08 are promising, but the use of other data sets may result in models that create excessive errors in areas of variability, which thus require more selective sampling.

Variable Density Grids

These DEMs attempt to maximise the advantages of the regular grid and sub-sampling, but in a constrained manner which reduces data redundancy, whilst limiting maximum error. This is accomplished with the use of sub-grids with cell sizes of 50, 100, 200 and 400 metres. Thus sub-sampling of the original 50 metre grid only occurs if the interpolated errors are within

tolerance thresholds. Since the model is based upon the regular grid, profile generation time is equivalent to that within a 50 metre dense grid, or faster if interpolations are made only at grid cell boundaries.

A comparison between the 10 metre (maximum error) grid and the sub-sampled 100 metre grid of ST08 (Table 9.2) illustrates some interesting points. The R.M.S. elevation error of the sub-sampled regular grid is considerably better than that of the variable grid, but R.M.S. path loss error is greater, even though both have an equivalent number of profiles within ± 6 dBs. This proves that even though more interpolated points within the variable grid are in error or have greater (tolerance-constrained) errors than the 100 metre grid, the effect on radio path loss is not as significant. It follows that small numbers of large elevation errors (ie. greater than ± 10 metres) have a significantly greater effect on radio path prediction than large numbers of smaller errors. This conclusion would therefore advocate the use of an elevation error-constrained DEM for best results at radio path loss prediction. The variable density grid conforms to this requirement, but the storage costs are still excessive. 95% of profiles are within radio path loss errors of ± 6 dBs in the 7.5 metre model, but with only a 54% storage reduction over the 50 metre grid.

Differential Altitude Grids

Data compressed regular grids or differential altitude grids attempt to make fuller use of the storage units that represent the grid elevations. Elevation differences are stored as coded increments of an average value, the magnitude of which is determined by a two or three bit code. Every original grid vertex is represented, but with many small elevation errors. The two-bit model has significantly worse elevation error than the regular grid models (eg. at 100 or 150 metres), but better path loss performance (Table 9.2). This is also true for the 15 metre variable grid. Anomalies such as this are achieved when a relatively large proportion of grid vertices are in error by more than ± 10 metres (2.14%). The effect of this is responsible for the small number of corrupted radio profiles, but the overall effect is reduced by a good overall surface performance. The largest errors occur in areas of steepest slope, such as valley sides, to the extent that the magnitude of elevation change between grid vertices cannot be efficiently modelled by a coded difference of limited magnitude. In many instances however, the slope of the valley is not as important as locating the point of highest obstruction, such as a peak or a ridge.

The conclusion for the variable density grid advocates the use of an elevation error constrained DEM, but the results for the differential altitude grid suggest that this might not be so important if the overall performance of the model is 'good'. In this instance, 'good' is defined as having small absolute errors (for example, less than two or three metres) for the majority of the surface. These two conclusions would therefore suggest that the most beneficial model is one

which combines the advantages of a maximum elevation error constrained model which removes data redundancy, but whose average error is relatively low or also constrained.

Huffman Encoded Grids

This regular grid DEM is data compressed in a similar manner to the differential altitude grid, but with one important difference. The encoded difference method reduces the redundancy created by using fixed storage units for the elevations, whilst Huffman coding not only achieves this, but also attempts to remove the data redundancy created by over-sampling of the original grid. This is accomplished using variable length codes which are adapted to the variability of the terrain. The prediction algorithm employed here, assumes that there is a local linear trend within the data, such that any deviations from this trend will be assigned longer, less probable codes, whilst uniform terrain is assigned shorter codes. Hence, data redundancy is identified and eliminated at a local level. This new approach enables error-free data compression of regular grid DEMs with storage savings of 80% to 90% with no radio path loss error.

Further compression is possible by banding grid heights into distinct classes, such that elevation error is constrained, whilst producing further storage savings of up to 50%. For both ST06 and ST08, over 95% of profiles can be estimated to within ± 6 dBs with total storage savings of 89% to 92%. These results are very encouraging, since they prove that the excessive data redundancy within regular grid DEMs can be removed. The results also highlight the significant degree of data redundancy within these data sets (80%+) and the disadvantages of using fixed size storage units (eg. two byte elevations).

Sub-Sampled Huffman Grids

Huffman encoding of sub-sampled regular grids produces vast storage savings (93% +) over the original 50 metre grid DEM, but the problems associated with sub-sampling still remain (ie. not error constrained; not selective; and too unpredictable). Huffman coding of regular grids at larger cell sizes does not produce the same degree of compression as with dense grids. A sparser sampling of elevations is likely to produce more deviations from the local linear predicted trend. As such, longer codes are required for these instances. Similarly, even denser grids (eg. at 30 metre intervals), such as the 7.5 minute quadrangle DEMs produced by the U.S.G.S. are likely to produce greater compression, due to their greater redundancy. This has been illustrated by Kidner and Smith (1991). In conclusion therefore, sub-sampled Huffman coding offers no significant improvements over error-constrained Huffman coding.

Variable Density Huffman Grids

A more selective, error-constrained approach to sub-sampled Huffman coding overcomes many of the above disadvantages and the relatively high storage costs of conventional variable

density grids. However, the degree of compression is only 60% to 65%, compared with over 80% for a regularly sampled 50 metre DEM. This supports the view that the greatest benefits of data compression are obtained for dense regular grids. Variable density Huffman grids do not offer the same compression ratios as for regular grids at similar levels of elevation and radio path loss performance.

Polynomial Surface Patches

Polynomial functions are the simplest and most flexible of all mathematical methods for terrain modelling. In many ways a polynomial surface patch DEM is very similar to the regular grid DEM. Instead of the grid cell being represented by a central elevation or four vertex elevations, a mathematical function describes the terrain within that cell. Whilst a regular grid has a dense sampling of vertices (eg. every 50 metres), the surface patch DEM is very sparse (eg. 500 metres to 1250 metres). For interpolation, the surface function can be used to approximate elevations without the need for costly search. As such, its computational efficiency is equivalent to that of the regular grid. By fitting the approximating functions to all original grid vertices, even quite variable terrain can be modelled efficiently.

The four byte polynomial coefficients are costly to store, so that at acceptable storage levels (ie. 75% saving), the performance of the polynomials is not as good. At such levels, the polynomials need to be of a low degree or the patches have to be large, with the result that the surface functions are not flexible enough to accurately model the terrain. The largest elevation errors occur in areas of steepest slope (ie. valleys), but the effect on radio path loss may not be too severe. However, critical obstructions or diffraction edges have a tendency to be displaced, which when combined with other small errors does have a major effect on radio path loss estimation. Another similarity with the regular grid is its inherent data redundancy. Polynomials of fixed degree applied to fixed-size patches are not adaptable to terrain variability. Maximum error cannot be constrained, creating significant radio path loss errors.

Adaptive Polynomial Surface Patches

The data redundancy of fixed degree polynomials is effectively removed by varying the degree of the approximating polynomial. Maximum absolute elevation error is therefore constrained, since any error above a certain tolerance will necessitate the use of a higher degree polynomial. The maximum degree needs to be predetermined, since in some instances the number of coefficients may become excessively large. For this model, the maximum polynomial degree is ten (66 coefficients). It is apparent however, that some terrain features are unsuitable for efficient polynomial modelling at certain scales (eg. patches greater than 1 km x 1 km). In particular, very steep slopes have a tendency to be smoothed or over-generalised, to such an extent that the maximum error tolerance cannot be attained, even with 10th degree polynomials. These errors are minimal, particularly for smaller patches, but it highlights the

limitations and inflexibility of mathematical modelling. The method is adaptable to terrain variability to a great extent, but uniform patch size undermines the flexibility of the approach. As a consequence, despite encouraging storage savings, radio path loss results do not have the required consistency. Most of the mathematical surface functions examined, including Fourier series, have a tendency to over-generalise or displace the key surface features needed for efficient path loss estimation.

Surface Patch Quadtree

The surface patch quadtree was designed to overcome many of the limitations of using mathematical surface models which are not fully constrained to the variability of terrain. The problems and more specifically, inflexibility associated with the polynomial surface patch DEMs are overcome by varying the patch or grid size, rather than the degree of the approximating function. An index of grid cells allows efficient storage and linear search of the quadtree. A simple ruled surface or bilinear surface function enables four grid vertices to represent the surface, rather than real coefficients of a mathematical function. As such, six bytes of storage per record can accurately represent a sub-grid of between 2×2 and 128×128 vertices, within a constrained error tolerance. The biggest drawback with the method is that surface representation is limited to $2^n \times 2^n$ grid values, where n is some positive integer. The DEMs currently supplied by the O.S. and USGS are constrained to national grid or latitude/longitude coordinates, such that implementation of the surface patch quadtree would require a major reorganisation of data. The performance of the surface patch quadtree is good in terms of storage, elevation error and radio path loss estimation. However, as the model becomes more error constrained to smaller tolerances, storage increases sharply. In this respect, the overall performance of the model is very similar to the sub-sampled regular grid.

Triangulated Irregular Network (TIN)

The TIN is the most popular alternative to the regular grid, because it is not only error constrained but is adaptable to terrain variability, such that there is little data redundancy. The vertices of the TIN are all critical surface points or features, such as peaks, pits, ridges, channels, breaklines or other information rich data. If the data have been directly sampled, and not derived from a regular grid, the TIN can be the most accurate form of digital elevation model.

A TIN can be represented in a number of different forms, depending upon whether the vertices, triangles or edges are considered as the primary entity of the data structure. Each of these representations may be expensive in terms of storage, in order to maintain the topological relationships that are required for applications that require search. For example, a vertex based TIN, as used in Chapter Eight, requires approximately ten times as much storage for every point derived from a regular grid DEM. A further associated problem is the

computational efficiency of such a TIN. Profiling is significantly faster in a triangle-based TIN, since the path can be traced from one triangle to another. However, this model requires even more storage than the vertex-based TIN. The storage requirements of such a DEM are too expensive to maintain as an efficient model in its current state. The maintenance of the explicit topology necessary to form the network cannot be considered viable in any storage efficient DEM to rival the regular grid.

Implicit TIN

Despite the storage overheads of maintaining a TIN, this form of DEM has many advantages over other models. A small subset of regular grid vertices can be adequate to represent an error-constrained DEM, using a TIN. For ST08, only 8.37% of the original 401x401 grid vertices are necessary to model the surface to within a maximum absolute error of 5 metres. The cost of storing all coordinates and pointers to represent the TIN increases this to over 83%. The Implicit TIN removes this excessive overhead by only storing the vertices in a compressed form based on a grid-cell data structure, such that all topology is derived at the application stage. Since the TIN is originally constructed to the error-constrained tolerance using a Delaunay triangulation algorithm, any local reconstruction of triangles (for example, as in profiling) will recreate the original topology and thus guarantee the error tolerance of the original model.

At the application stage, the triangulation is reconstructed in the spatial window of interest for each profile. A sufficient boundary for this will allow other profiles in this vicinity to be interpolated without any further TIN reconstruction. Profile retrieval time is therefore directly related to the number of TIN vertices or indirectly to the variability of the terrain. As such, the Implicit TIN allows a high degree of accuracy in terms of elevation (100% of points within ± 5 metres) and radio path loss (> 95% of profiles within ± 6 dBs), with over 90% storage savings. The Implicit TIN is a better indicator than the more conventional explicit TIN of the advantages to be gained from using an irregular point representation.

9.4 Research Summary

The performances of the digital elevation models described in this study have all been compared to that of the dense 50 metre regular grid DEM. This model has been used as the 'real world' model for the surfaces represented by ST06 and ST08. It should always be remembered that there are inherent errors in using such a model, due to its limited accuracy, lack of detailed terrain coverage information, such as vegetation and buildings, and inflexible sampling density. All these deficiencies will create radio path loss errors which cannot be accurately determined. Further radio path loss errors result from the use of the computationally efficient bilinear interpolation, rather than the more accurate 16-term bicubic interpolation. Hence all DEM results have a one or two percent margin of error in terms of radio path profiles within ± 6

dBs, when compared to the 'real world'.

One of the most important criteria in the determination of a DEM for radio path loss estimation is storage efficiency. As such, the dense regular grid DEM is unsatisfactory, because of its major disadvantage of data redundancy due to inefficient sampling and inability to model terrain variability. The alternative DEM must significantly improve upon the storage overheads of the regular grid to be considered worthwhile. The general consensus as to '*what is an acceptable level of improvement in storage efficiency*' has, from the overall results, been determined as 25% of the storage cost of the original 50 metre grid. This figure has been ascertained by considering the storage saving necessary to compensate for a possible increase in elevation error, radio path loss error and profile retrieval time. The levels of performance of the secondary criteria, such as elevation error, radio path loss error and computational efficiency are more open to question or 'trade-off'. Storage costs however, cannot be compromised.

At the outset of this study, it was originally thought that elevation error would be the major criterion against which DEM performance for storage-efficient models would be examined. It is natural to assume that an increase in average elevation error will lead to a subsequent increase in radio path loss error, for any DEM. However, this is a misconception which is dangerous to accept unconditionally. In many instances this premise is true, but more importantly the conditions in which it is false need further examination.

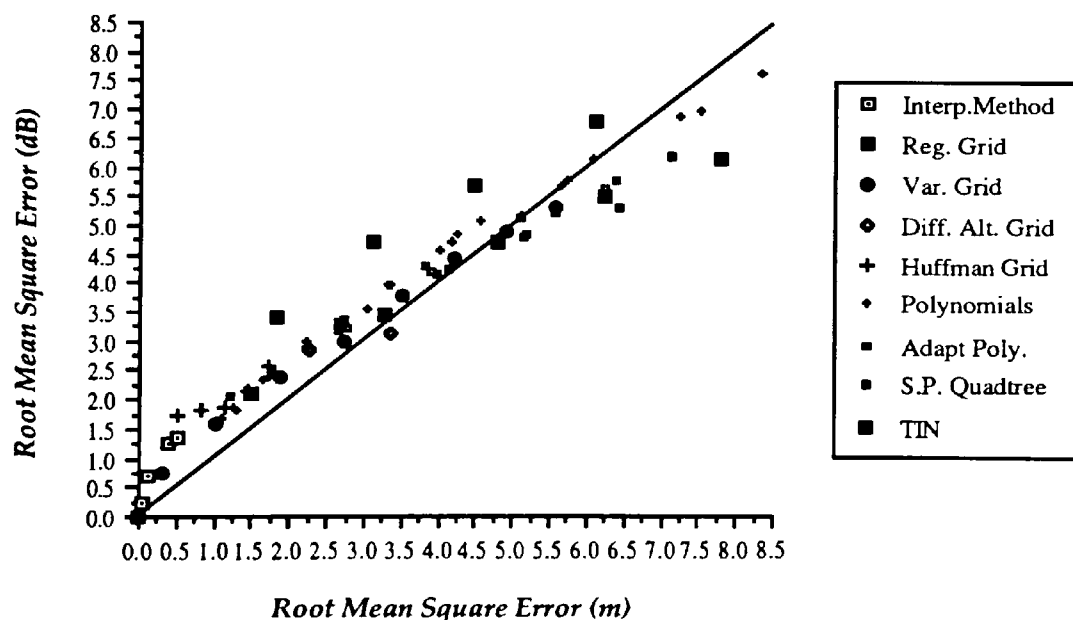


Figure 9.4 - Graph of Elevation Error (RMSE in metres) v. Radio Path Loss Error (RMSE in dBs).

Figure 9.4 illustrates the relationship between RMS elevation error and RMS radio path loss error for the data models applied to ST08. This graph also includes the results for choice of

grid interpolation method (ie. linear, biquadratic, bicubic, etc.). It is noticeable however, that there may be a large deviation in RMS radio path loss error for models of similar RMS elevation error. The DEM which best illustrates this phenomenon is the differential altitude grid. In Tables 9.1 and 9.2 (and Figure 9.4), it can be seen that this DEM produces significantly better radio path loss results than the regular and sub-sampled grids, despite comparatively worse elevation errors. On closer inspection, this was discovered to be as a result of the position of the elevation errors. The magnitude of elevation differences between grid nodes cannot always be modelled with fixed-length codes, so that large errors will occur in areas of steep slope, such as valleys. These errors significantly distort the overall average elevation error, despite a generally good overall performance. However, these errors are not likely to create excessive radio path loss errors (apart from some increased reflection losses), unless the critical diffraction edges, such as peaks and ridges are over-generalised. The local nature of the differential altitude grids ensures that this is uncommon. Hence unconstrained elevation error is not necessarily a major disadvantage of any particular DEM. More important is where these errors occur.

The general results for unconstrained DEMs suggests that most models produce correspondingly large radio path losses, due to the unpredictable nature of elevation errors. For example, polynomial models also create large elevation errors in regions of steep slope, but despite some good path loss estimates for profiles in these regions, other profiles suffer due to a displacement of peaks and ridges. In general therefore, unconstrained elevation errors are not recommended in any alternative DEM for radio path loss prediction. The differential altitude grid may disprove this to some extent, but the fidelity of critical features such as peaks and ridges cannot be guaranteed to be accurately represented. As such, there will always be a degree of unpredictability associated with any such model.

An alternative DEM should therefore be error-constrained to some maximum absolute elevation tolerance and should provide an overall consistent representation with small average errors. The important attributes of such a model should also accurately represent the critical terrain features needed for radio path loss prediction. These are not only the possible diffraction edges, such as ridges and peaks, but also to a lesser extent, the degree of slope in areas of variable terrain. Slope affects radio path loss in terms of reflection losses and siting losses near the transmitter and receiver.

As already stated, the radio path loss algorithm used in this study is very sensitive to small elevation errors in certain circumstances. The majority of radio path loss errors are caused by an obstruction or diffraction edge being undetected or inserted, or misplaced by a few profile interpolated points (eg. 50 to 250 metres). Some of these errors are compensated for by an increase or decrease in magnitude of the diffraction losses of other obstructions, but more than

one such error in any profile will create a large path loss error. Such occurrences are the major cause of most profiles which have radio path loss errors of greater than ± 6 dBs. The number of such profiles has a tendency to be greater at higher frequencies (ie. 1400 to 1800 MHz). Some profiles are particularly sensitive at lower frequencies (ie. 200 MHz) to small hills in the vicinity of the transmitter or receiver, creating large siting losses, despite unobstructed 'line-of-sight' profiles. Errors in the degree of slope near the transmitter and receiver may also create such siting losses at 200 MHz. In general therefore, these results suggest that the most consistent radio path loss estimates are achieved at frequencies of 400 to 900 MHz, and 600 MHz in particular.

The auxiliary criterion on which the choice of an alternative DEM is made, is computational efficiency or profile retrieval time. However, it is apparent that no alternative, storage efficient and error-constrained DEM will perform as efficiently as bilinear interpolation within a dense regular grid DEM. The 'trade-off' in achieving storage efficiency is usually a resulting decrease in computational efficiency. More commonly, decreasing the degree of topology of a data structure (storage efficiency), will remove many of the inherent search facilities of a DEM. This is particularly apparent with the Implicit TIN. As such, profile interpolation time needs to be examined in a wider context. The computational cost of predicting radio path loss within a regular grid DEM is approximately 15 times greater than estimating the profile heights, and yet is measured in terms of milliseconds, rather than seconds. Hence, the average time taken for any DEM to locate a suitable siting of a transmitter or receiver will be relatively small. In the worst case of having to uncompress an error-free Huffman-encoded DEM, the maximum 'one-off' overhead is five seconds of CPU time.

Even though computational efficiency is an important criterion, it is perhaps the most flexible of all the factors in determining an alternative DEM. Whilst storage efficiency is a necessity, the user is left with a choice between accurate radio path loss prediction or computational efficiency. However, the relationship between radio path loss and elevation errors is not consistent or predictable. A relaxation in the criterion that 95% of profiles are within ± 6 dBs is not viable, since it is accomplished at a cost of increasing elevation errors. For some data sets, this may not significantly affect the overall path loss errors, but for other data sets, these path loss errors may be intolerable. The importance of this should not be under-estimated, since radio path loss calculation is the major application of the DEM. In this respect, it can be concluded that computational efficiency is the most flexible of all requirements. In the broader context of other applications, time may not be as important as storage efficiency or error performance.

9.5 Final Conclusions

In summary therefore, the requirements of a DEM for radio path loss estimation in a mobile communications network are that the model should:

- (1) be storage efficient (ie. 75% storage saving over the regular grid DEM);
- (2) be error constrained (ie. 100% of points within ± 10 metres);
- (3) produce a small overall average elevation error, if any;
- (4) preserve critical terrain characteristics such as ridges, peaks and slopes; and
- (5) produce a good performance in terms of radio path loss estimation (ie. 95% of profiles within ± 6 dBs).

The models which fit these criteria can be identified from Table 9.3.

The two models which give the best overall performance are the Implicit TIN and the Huffman-encoded grid DEM. If an error-free representation of the original grid DEM is required (ie. a data compressed model), then the linear prediction algorithm combined with the Huffman encoded DEM will achieve this with storage savings of 80% or more. Preliminary results from the analysis of 43 O.S. grid DEMs covering South Wales and Southern England, suggest in all cases that error-free compression of at least 78% can be achieved (Appendix D). This storage saving can be increased by a further 50%, if constrained elevation errors of up to 2.5 metres are allowed. Thus error-constrained Huffman-encoded DEMs will allow the above criterion to be achieved with storage savings of approximately 90% or more. Huffman coding identifies and removes the redundancy within a regular grid DEM at a local level.

The Implicit TIN can improve upon this compression ratio for the equivalent data sets (ie. ST06 and ST08), at the cost of having larger absolute elevation errors. The Implicit TIN and TINs in general, illustrate the data redundancy within regular grid DEMs. This data redundancy is removed in a totally different approach from Huffman coding. Insignificant or data redundant vertices are globally removed, together with information-rich points. The critical points and features are iteratively reintroduced at a progressively more local level, as the error tolerance becomes more constrained. As a result, storage savings in excess of 90% can also be achieved over the regular grid DEM. However, the complexity of implementing a TIN or Implicit TIN will require a more significant degree of understanding than for a grid-based DEM, such as the Huffman-encoded grid.

In conclusion, this research has shown that the Implicit TIN and the Huffman-encoded grid are both acceptable as viable alternatives to the regular grid DEM. In both instances, satisfactory results can be achieved with storage savings of 90% or more. As the cost of data storage falls, the user will have to decide whether such storage savings are worth the loss of flexibility. If computational efficiency is considered an overriding criterion, the regular or sub-sampled grid

DEM should be used. However, the disadvantage of restoring data-compressed DEMs into more flexible data structures can often be overcome with the application of parallel processing techniques. The Delaunay triangulation of the Implicit TIN is unique, such that it may be reconstructed in parallel (Ware & Kidner, 1991). Similarly, a regular grid DEM can be reconstructed in parallel from the Huffman-encoded DEM, provided the global look-up table of codewords is valid over each partition of the grid. With the development of parallel processing, the potential use of high compression for long term storage and temporary data structures, such as the regular grid or TIN, would be of great benefit to many users.

The future of digital terrain modelling will no doubt see a vast increase in the amount of data that becomes available. As the Ordnance Survey nears completion of the national 50 metre regular grid data sets, the need for more detailed and accurate data is already apparent for applications such as radio path loss calculation. The growth in GIS will hopefully ensure that this demand is met. Developments in GIS suggest that there will always be a requirement for storage-efficient DEMs, despite the likelihood of less expensive storage and faster machines. For GIS in general, the growth in data is expected to exceed the fall in computing costs. This has already been proven by the vast quantities of data obtained from satellite imagery. Hence, compression of spatial data, including digital elevation data, will be a major issue for the future.

Cited References

- Abel, D.J. & Smith, J.L. (1983)** "A Data Structure and Algorithm Based on a Linear Key for a Rectangle Retrieval Problem", *Computer Vision, Graphics, and Image Processing*, Vol.24, pp.1-13.
- Abel, D.J. (1985)** "Some Elemental Operations on Linear Quadrees for Geographic Information Systems", *The Computer Journal*, Vol.28, No.1, Jan., pp.73-77.
- Ackermann, F. (1978)** "Experimental Investigation into the Accuracy of Contouring from DTM", *Photogrammetric Engineering and Remote Sensing*, Vol.44, No.12, Dec., pp.1537-1548.
- Ackeret, J.R. (1989)** "Digital Topographic Data Requirements Analysis for the Mobile Subscriber Equipment (MSE) System", *ASPRS/ACSM*, Vol.4, GIS/LIS, Apr. 2-7, pp.214-235.
- Akima, H. (1974a)** "A Method of Bivariate Interpolation and Fitting Based on Smooth Local Procedures", *Communications of the ACM*, Vol.17, No.1, Jan., pp.18-20.
- Akima, H. (1974b)** "Bivariate Interpolation and Smooth Surface Fitting Based on Local Procedures", *Communications of the ACM*, Vol.17, No.1, Jan., pp.26-31.
- Allder, W.R., Caruso, V.M., Pearsall, R.A. & Troup, M.I. (1982)** "An overview of Digital Elevation Model Production at the United States Geological Survey", *Proc. of Auto Carto V*, Aug. 22-28, pp.23-32.
- Ayeni, O.O. (1978)** "Automated Digital Terrain Models", *Proc. of the Digital Terrain Models (DTM) Symposium*, ASP/ACSM, May 9-11, pp.275-306.
- Baker, P.W., Davies, P.G. & Haley, R. (1983)** "Propagation Characteristics and Prediction Techniques Relevant to Land Mobile Applications", *Proc. of 2nd International Conference on Radio Spectrum Conservation Techniques*, IEE, Birmingham, England, Sept.6-8, pp.61-65.
- Balce, A.E. (1987)** "Determination of Optimum Sampling Interval in Grid Digital Elevation Models (DEM) Data Acquisition", *Photogrammetric Engineering and Remote Sensing*, Vol.53, No.3, Mar., pp.323-330.
- Balch, S.J. & Thompson, G.T. (1988)** "An Efficient Algorithm for Polynomial Curve Fitting", *Computers and Geosciences*, Vol.14, No.5, pp.547-556.
- Balch, S.J. & Thompson, G.T. (1989)** "An Efficient Algorithm for Polynomial Surface Fitting", *Computers and Geosciences*, Vol.15, No.1, pp.107-119.
- Barnhill, R.E. (1977)** "Representation and Approximation of Surfaces", in *Mathematical Software III*, *Proc. of Symposium*, University of Wisconsin - Madison, Editor J.R. Rice, Academic Press, Mar. 28-30, pp.69-120.
- Barrera, R. & Vazquez, A.M. (1984)** "A Hierarchical Method for Representing Terrain Relief", *IEEE*, pp.87-92.
- Boehm, B.M. (1967)** "Tabular Representation of Multivariate Functions with Applications to Topographic Modelling", *Proc. of the 22nd National Conference of ACM*, pp.403-415.
- Bouillé, F. (1978)** "Structuring Cartographic Data and Spatial Processes with the Hypergraph-Based Data Structure", *First International Advanced Study Symposium on Topological Data Structures for Geographic Information Systems*, *Harvard Papers on GIS*, (Geoffrey Dutton, Editor), Vol.5, Data Structures: Surficial and Multi-Dimensional, 16 pages.

- Bowyer, A. (1981)** *"Computing Dirichlet Tessellations"*, The Computer Journal, Vol.24, No.2, Feb., pp.162-166.
- Burrough, P.A. (1986)** *Principles of Geographic Information Systems for Land Resources Assessment*, Oxford University Press, 193 pages.
- Burton, W. (1978)** *"Efficient Retrieval of Geographical Information on the Basis of Location"*, First International Advanced Study Symposium on Topological Data Structures for Geographic Information Systems, Harvard Papers on GIS, (Geoffrey Dutton, Editor), Vol.6, Spatial Algorithms: Efficiency in Theory and Practice, 23 pages.
- Carter, J.R. (1988)** *"Digital Representation of Topographic Surfaces"*, Photogrammetric Engineering and Remote Sensing, Vol.54, No.11, Nov., pp.1577-1580.
- Catlow, D.R. (1986)** *"The Multi-Disciplinary Applications of DEMs"*, Proc. of Auto Carto London, Vol.1, Sep. 14-19, pp.447-454.
- Cebrian, J.A., Mower, J.E. & Mark, D.M. (1985)** *"Analysis and Display of Digital Elevation Models within a Quadtree-Based Geographic Information System"*, Proc. of Auto Carto VII, Mar.11-14, pp.55-64.
- Chen, Zi-Tan & Tobler, W. (1986)** *"Quadtree Representations of Digital Terrain"*, Proc. of Auto Carto London, Vol.1, Sep. 14-19, pp.475-484.
- Chen, Zi-Tan & Guevara, J.A. (1987)** *"Systematic Selection of Very Important Points (VIP) from Digital Terrain Models for Constructing Triangulated Irregular Networks"*, Proc. of Auto Carto VIII, Mar. 29-Apr. 3, pp.50-56.
- Chorley, Lord (1987)** *Handling Geographic Information*, Report of the Committee of Enquiry chaired by Lord Chorley, Department of the Environment, H.M.S.O., London, 208 pages.
- Christensen, A.H.J. (1987)** *"Fitting a Triangulation to Contour Lines"*, Proc. of Auto Carto VIII, Mar. 29-Apr. 3, pp.57-67.
- Clarke, A.L., Gruen, A. & Loon, J.C. (1982)** *"The Application of Contour Data for Generating High Fidelity Grid Digital Elevation Models"*, Proc. of Auto Carto V, Aug. 22-28, pp.213-222.
- Clarke, K.C. (1985)** *"Strategies for Spatial Data Compression"*, Proc. of Auto Carto VII, Mar.11-14, pp.97-107.
- Clarke, K.C. (1990)** *Analytical and Computer Cartography*, Prentice Hall, Englewood Cliffs, New Jersey, 290 pages.
- Cole, G., MacInnes, S. & Miller, J. (1990)** *"Conversion of Contoured Topography to Digital Terrain Data"*, Computers & Geosciences, Vol.16, No.1, Jan., pp.101-109.
- Collins, G. (1989)** *"A Fast Recursive Parallel Algorithm for Triangulation of Unordered DTM's"*, Cartography, Vol.18, No.1, June, pp.21-25.
- Collins, S.H. (1981)** *"Algorithms for Dense Digital Terrain Models"*, Photogrammetric Engineering and Remote Sensing, Vol. 47, No.1, Jan., pp.71-76.
- Coppock, T. & Anderson, E. (1987)** Editorial Review, Int. Journal of Geographical Information Systems, Vol.1, No.1, pp.3-11.
- Correc, Y. & Chapuis, E. (1987)** *"Fast Computation of Delaunay Triangulations"*, Advanced Eng. Software, Vol.9, No.2, pp.77-83.

- Cliff, A.D., Haggett, P., Ord, J.K., Bassett, K. & Davies, R. (1975) *Elements of Spatial Structure - A Quantitative Approach*, Cambridge University Press, Cambridge, England.
- Crain, I.K. (1970) "Computer Interpolation and Contouring of Two-Dimensional Data: A Review", *Geoexploration*, Vol.8, pp.71-86.
- Dadson, C.E. (1979) "Radio Network and Radio Link Surveys Derived by Computer from a Terrain Data Base", AGARD Conference Proc. No.269: Terrain Profiles and Electromagnetic Wave Propagation, Spatind, Norway, Sept.10-14, pp.25.1-25.17.
- Daly, R. (1989) "Scale-Free Data Structures for G.I.S.", Paper Presented at the Conference on Managing Geographical Data and Databases, Lancaster.
- Date, C.J. (1983) *An Introduction to Database Systems*, Vol.11, Addison-Wesley, Reading, Mass.
- Davis, D.M., Downing, J. & Zoraster, S. (1982) "Algorithms for Digital Terrain Data Modeling", Technical Report DAAK70-80-C-0248, U.S. Army Engineer Topographic Laboratories, Fort Belvoir, Virginia.
- Davis, J.C. (1986) *Statistics and Data Analysis in Geology*, 2nd Edition, John Wiley & Sons, New York, 550 pages.
- De Floriani, L., Falcidieno, B. & Pienovi, C. (1983) "A Delaunay-Based Method for Surface Approximation", *Proc. of Eurographics '83*, Univ. of Zagreb, Aug. 31-Sept. 2, pp.333-350.
- De Floriani, L., Falcidieno, B., Nagy, G. & Pienovi, C. (1984) "A Hierarchical Structure For Surface Approximation", *Computer & Graphics*, Vol.8, No.2, pp.183-193.
- De Floriani, L., Falcidieno, B., Pienovi, C. & Nagy, G. (1985a) "Efficient Selection, Storage, and Retrieval of Irregularly Distributed Elevation Data", *Computers and Geosciences*, Vol.11, No.6, July, pp.667-673.
- De Floriani, L., Falcidieno, B. & Pienovi, C. (1985b) "Delaunay-Based Representation of Surfaces Defined Over Arbitrarily Shaped Domains", *Computer Vision, Graphics And Image Processing*, Vol.32, pp.127-140.
- De Floriani, L. (1987) "Surface Representations Based on Triangular Grids", *The Visual Computer*, Vol.3, No.1, Feb., pp.27-50.
- De Floriani, L. & Puppo, E. (1988) "Constrained Delaunay Triangulation for Multiresolution Surface Description", *Proc. of the 9th International Conference on Pattern Recognition*, Rome, Nov.14-17, pp.566-569. (Reprinted by IEEE Computer Society Press).
- De Floriani, L. (1989) "A Pyramidal Data Structure for Triangle-Based Surface Description", *IEEE Computer Graphics and Applications*, pp.67-78.
- Delfiner, P. & Delhomme, J.P. (1975) "Optimum Interpolation by Kriging", in *Display and Analysis of Spatial Data*, (NATO Advanced Study Institute, 1973), Editors J.C. Davis & M.J. McCullagh, John Wiley and Sons, London, pp.96-114.
- Devereux, B.J. (1985) "The Construction of Digital Terrain Models on Small Computers", *Computers and Geosciences*, Vol.11, No.6, pp.713-724.
- Douglas, D.H. (1986) "Experiments to Locate Ridges and Channels to Create a New Type of Digital Elevation Model", *Cartographica*, Vol.23, No.4, Dec., pp.29-61.
- Doyle, F.J. (1978) "Digital Terrain Models: An Overview", *Photogrammetric Engineering and Remote Sensing*, Vol.44, No.12, Dec., pp.1481-1485.

- Dutton, G. (1983) *"Efficient Encoding of Gridded Surfaces"*, Spatial Algorithms for Processing Land Data with a Microcomputer, Report of Organisation of the Lincoln Institute for Land Policy, Cambridge, Massachusetts, USA, pp.23-62.
- Dutton, G. (1984) *"Geodesic Modelling of Planetary Relief"*, Cartographica, Vol.21, pp.188-207.
- Dutton, G. (1989) *"Planetary Modelling via Hierarchical Tessellation"*, Proc. of Auto Carto 9, Tech. Papers of ACSM/ASPRS, Baltimore, Maryland, Apr.2-7, pp.462-471.
- Dutton, G. (1990) *"Locational Properties of Quaternary Triangular Meshes"*, Proc. of 4th Int. Symposium on Spatial Data Handling, Zurich, July 23-27, Vol.2, pp.901-910.
- Dutton, G. (1991) *"Improving Spatial Analysis in GIS Environments"*, Proc. of Auto Carto 10, Tech. Papers of ACSM/ASPRS, Vol.6, Baltimore, Maryland, Mar.25-28, pp.168-185.
- Dwyer, R.A. (1987) *"A Faster Divide-and-Conquer Algorithm for Constructing Delaunay Triangulations"*, Algorithmica, Vol.2, pp.137-151.
- Edwards, R. & Durkin, J. (1969) *"Computer Prediction of Service Areas for v.h.f. Mobile Radio Networks"*, Proc. of the IEEE, Vol.116, Electronics, No.9, Sep., pp.1493-1500.
- ElGindy, H. (1986) *"An Optimal Speed-Up Parallel Algorithm for Triangulating Simplicial Point Sets in Space"*, International Journal of Parallel Programming, Vol. 15, No.5, pp.389-398.
- ElGindy, H. (1990) *"Optimal Parallel Algorithms for Updating Planar Triangulations"*, in Proc. of the 4th Int. Symposium on Spatial Data Handling, Zurich, July 23-27, Vol.1, pp.200-208.
- Fisher, P.F. (1991) *"Simulation of the Uncertainty of a Viewshed"*, Proc. of Auto Carto 10, Tech. Papers of ACSM/ASPRS, Vol.6, Baltimore, Maryland, Mar.25-28, pp.205-218.
- Fowler, R.J. & Little, J.J. (1979) *"Automatic Extraction of Irregular Network Digital Terrain Models"*, Computer Graphics, Vol.13, No.1, Jan., pp.199-207.
- Franke, R. (1982) *"Scattered Data Interpolation: Tests of Some Methods"*, Mathematics of Computation, Vol.38, No.157, Jan., pp.181-200.
- Frederiksen, P., Jacobi, O. & Kubik, K. (1985) *"A Review of Current Trends in Terrain Modelling"*, The ITC Journal, No.2, pp.101-106.
- Gargantini, I. (1982) *"An Effective Way to Represent Quadtrees"*, Communications of the ACM, Vol.25, No.12, Dec., pp.905-910.
- Gillman, D.W. (1985) *"Triangulations for Rubber-Sheeting"*, Proc. of Auto Carto VII, Mar. 11-14, pp.191-199.
- Gold, C.M. (1978) *"The Practical Generation and Use of Geographic Triangular Element Data Structures"*, First International Advanced Study Symposium on Topological Data Structures for Geographic Information Systems, Harvard Papers on GIS, (Geoffrey Dutton, Editor), Vol.5, Data Structures: Surficial and Multi-Dimensional, 18 pages.
- Gold, C.M. (1979) *"Triangulation-Based Terrain Modelling - Where are we now ?"*, Proc. of Auto Carto IV, Vol.2, Nov. 4-8, pp.104-111.
- Gold, C.M. (1987) *"Spatial Ordering of Voronoi Networks and Their Use in Terrain Data Base Management"*, Proc. of Auto Carto VIII, Mar. 29-Apr. 3, pp.185-194.

- Gold, C.M. & Cormack, S. (1987) *"Spatially Ordered Networks and Topographic Reconstructions"*, *Int. Journal of Geographical Information Systems*, Vol.1, No.2, pp.137-148.
- Goodchild, M.F. & Yang Shiren (1990) *"A Hierarchical Data Structure for Global Geographic Information Systems"*, *Proc. of 4th Int. Symposium on Spatial Data Handling*, Zurich, July 23-27, Vol.2, pp.911-917.
- Grant, F. (1957) *"A Problem in the Analysis of Geophysical Data"*, *Geophysics*, Vol.22, No.2, Apr., pp.309-344.
- Green, P.J. & Sibson, R. (1978) *"Computing Dirichlet Tessellations in the Plane"*, *The Computer Journal*, Vol.21, No.2, Feb., pp.168-173.
- Greenlee, D.D. (1987) *"Raster and Vector Processing for Scanned Linework"*, *Proc. of Auto Carto VIII*, Mar. 29-Apr. 3, pp.640-649.
- Grosskopf, R. (1987) *"Comparison of Different Methods for the Prediction of the Field Strength in the VHF Range"*, *IEEE Trans. on Antennas and Propagation*, Vol.AP-35, No.7, July, pp.852-859.
- Hall, C. (1977) *"An Algorithm for the Production of an Isometric Projection of a Three-Dimensional Surface"*, NPL Report, NAC75.
- Hannah, M.J. (1981) *"Error Detection and Correction in Digital Terrain Models"*, *Photogrammetric Engineering and Remote Sensing*, Vol.47, No.1, Jan., pp.63-69.
- Harbaugh, J.W. & Merriam, D.F. (1968) *Computer Applications in Stratigraphic Analysis*, John Wiley and Sons, 282 pages
- Hardy, R.L. (1971) *"Multiquadric Equations of Topography and Other Irregular Surfaces"*, *Journal of Geophysical Research*, Vol.76, No.8, Mar., pp.1905-1915.
- Hardy, R.L. (1975) *"Research Results in the Application of Multiquadric Equations to Surveying and Mapping Problems"*, *Surveying and Mapping*, Dec., pp.321-332.
- Hayes, J.G. & Halliday, J. (1974) *"The Least-Squares Fitting of Cubic Spline Surfaces to General Data Sets"*, *Journal of the IMA*, Vol.14, pp.89-103.
- Heap, B.R. & Pink, M.G. (1969) *"Three Contouring Algorithms"*, NPL Report, DNAM81.
- Held, G. (1987) *Data Compression - Techniques and Applications*, Hardware and Software Considerations, 2nd Edition, John Wiley & Sons, England, 206 pages.
- Heller, M. (1986) *"Triangulation and Interpolation of Surfaces"*, in Chapter 3 of *A Selected Bibliography on Spatial Data Handling: Data Structures, Generalization and Three-Dimensional Mapping* (Editors: R.Sieber & K.Brassel), *Geoprocessing Series*, Vol.6, Dept. of Geography, Univ. of Zurich, pp.36-45.
- Heller, M. (1990) *"Triangulation Algorithms for Adaptive Terrain Modelling"*, *Proc. of the 4th Int. Symposium on Spatial Data Handling*, Vol.1, Zurich, July 23-27, pp.163-174.
- Hodgson, M.E. (1989) *"Searching Methods for Rapid Grid Interpolation"*, *Professional Geographer*, Vol.41, No.1, pp.51-61.
- Huffman, D.A. (1952) *"A Method for the Construction of Minimum-Redundancy Codes"*, *Proceedings of IRE*, Vol.40, Sept., pp.1098-1101. (Reprinted in *Key Papers in the Development of Information Theory*, Edited by D.Slepian, IEE Press, New York, 1974, pp.47-50).
- Jancaitis, J.R. & Junkins, J.L. (1973) *"Modelling Irregular Surfaces"*, *Photogrammetric*

- Engineering and Remote Sensing, Vol.39, No.4, Apr., pp.413-420.
- Jancaitis, J.R. & Junkins, J.L. (1973) *"Mathematical Techniques for Automated Cartography"*, Technical Report DAAK02-72-C-0256, U.S. Army Engineer Topographic Laboratories, Fort Belvoir, Virginia.
- Jancaitis, J.R. (1978) *"Elevation Data Compaction by Polynomial Modelling"*, Technical Report, U.S. Army Engineer Topographic Laboratories, Fort Belvoir, Virginia, 45 pages.
- Jenson, S.K. (1985) *"Automated Derivation of Hydrologic Basin Characteristics from Digital Elevation Model Data"*, Proc. of Auto Carto VII, Mar. 11-14, pp.301-310.
- Jenson, S.K. & Domingue, J.O. (1988) *"Extracting Topographic Structure from Digital Elevation Data for Geographic Information System Analysis"*, Photogrammetric Engineering and Remote Sensing, Vol.54, No.11, Nov., pp.1593-1600.
- Jones, D.M. & Knight, D.G. (1987) *"Spectral Resource Requirements of a Fielded Network: The Effect of Errors in End Positions and Terrain Heights on Path Loss Prediction"*, R.S.R.E. Report, Dept. of Mathematics and Computing, The Polytechnic of Wales.
- Jones, D.M. & Knight, D.G. (1988) *"Spectral Resource Requirements of a Fielded Network: Identifying Sensitive Path Loss Profiles"*, R.S.R.E. Report, Dept. of Mathematics and Computing, The Polytechnic of Wales.
- Junkins, J.L., Miller, J.R. & Jancaitis, J.R. (1973) *"A Weighting Function Approach to Modelling of Irregular Surfaces"*, Journal of Geophysical Research, Vol.78, No.11, Apr., pp.1794-1803.
- Kao, T., Mount, D. & Saalfeld, A. (1991) *"Dynamic Maintenance of Delaunay Triangulations"*, Proc. of Auto Carto 10, Tech. Papers of ACSM/ASPRS, Vol.6, Baltimore, Maryland, Mar.25-28, pp.219-233.
- Kennie, T.J.M. (1987) *"Software Packages for Terrain Modelling Applications"*, Proc. of Terrain Modelling in Surveying and Civil Engineering, Univ. of Surrey, Apr. 7-9, Univ. of Glasgow, Sep. 1-3.
- Kennie, T.J.M. & McLaren, R.A. (1988) *"Modelling for Digital Terrain and Landscape Visualisation"*, Photogrammetric Record, Vol.12, No.72, Oct., pp.711-745.
- Ketteman, M.R. (1987) *"The Applications of Terrain Modelling for Mine Surveying"*, Proc. of Terrain Modelling in Surveying and Civil Engineering, Univ. of Surrey, Apr. 7-9, Univ. of Glasgow, Sep. 1-3, 14 pages.
- Kidner, D.B., Jones, C.B., Knight, D.G. & Smith, D.H. (1990) *"Digital Terrain Models for Radio Path Profiles"*, Proc. of the 4th Int. Symposium on Spatial Data Handling, Vol.1, Zurich, July 23-27, pp.240-249.
- Kidner, D.B. & Jones, C.B. (1991) *"Implicit Triangulations for Large Terrain Databases"*, Proceedings of 2nd European Conference on GIS, Brussels, Apr. 2-5, Vol.1, pp.537-546.
- Kidner, D.B. & Smith, D.H. (1991) *"Compression of Digital Elevation Models by Huffman Coding"*, Paper Submitted to Computers and Geosciences, August 1991.
- Knuth, D.E. (1973) *The Art of Computer Programming, Vol.1, Fundamental Algorithms*, Addison-Wesley, Reading, Mass., USA.
- Kostli, A. & Wild, E. (1984) *"A Digital Elevation Model Featuring Varying Grid Size"*, Paper of the ISPRS Commission III, Rio de Janeiro, Brazil, pp.1130-1138.
- Kostli, A. & Sigle, M. (1986) *"The Random Access Data Structure of the DTM Program SCOP"*,

- Proc. of Mapping from Modern Imagery Symposium, ISPRS and The Remote Sensing Society, Edinburgh, Scotland, Sep. 8-12, pp.45-52.
- Krumbein, W.C. & Graybill, F.A. (1965)** *An Introduction to Statistical Models in Geology*, International Series in the Earth Sciences, McGraw-Hill, 475 pages.
- Kumler, M.P. (1990)** *"A Quantitative Comparison of Regular and Irregular Digital Terrain Models"*, Proc. of GIS/LIS '90, Anaheim, California, pp.255-263.
- Lam, N.S. (1983)** *"Spatial Interpolation Methods: A Review"*, The American Cartographer, Vol.10, No.2, pp.129-149.
- Leberl, F. (1973)** *"Interpolation in Square Grid DTM"*, ITC Journal
- Lee, D.T. & Schachter, B.J. (1980)** *"Two Algorithms for Constructing a Delaunay Triangulation"*, Int. Journal of Computer and Information Sciences, Vol.9, No.3, Feb., pp.219-242.
- Lee, J. (1989)** *"A Drop Heuristic Conversion Method for Extracting Irregular Networks for Digital Elevation Models"*, Proc. of GIS/LIS '89, Orlando, Florida, Nov., Vol.1, pp.30-39.
- Lee, J. (1991)** *"Assessing the Existing Methods for TIN Extraction"*, Proc. of ACSM/ASPRS, Baltimore, Maryland, Mar.25-28, Vol.2, Cartography and GIS/LIS, pp.194-203.
- Leifer, L.A. & Mark, D.M. (1987)** *"Recursive Approximation of Topographic Data Using Quadrees and Orthogonal Polynomials"*, Proc. of Auto Carto VIII, Mar. 29-Apr.3, pp.650-659.
- Lelewer, D.A. & Hirschberg, D.S. (1987)** *"Data Compression"*, ACM Computing Surveys, Vol.19, No.3, Sept., pp.261-296.
- L'Eplattenier, R & Sieber, R. (1986)** *"General Aspects of 3-D Mapping"*, in Chapter 4 of A Selected Bibliography on Spatial Data Handling: Data Structures, Generalization and Three-Dimensional Mapping (Editors: R.Sieber & K.Brassel), Geoprocessing Series, Vol.6, Dept. of Geography, Univ. of Zurich, pp.46-58.
- Ley, R.G. (1986)** *"Accuracy Assessment of Digital Terrain Models"*, Proc. of Auto Carto London, Vol.1, Sep. 14-19, pp.455-464.
- Little, J.J. (1978)** *"Strategies for Interfacing Geographic Information Systems"*, First International Advanced Study Symposium on Topological Data Structures for Geographic Information Systems, Harvard Papers on GIS, (Geoffrey Dutton, Editor), Vol.5, Data Structures: Surficial and Multi-Dimensional, 11 pages.
- Loon, J.C. (1984)** *"Computer Assisted Generalization of the Relief Continuum on a Topographic Map"*, Technical Papers of the 12th Conference of International Cartographical Association: 7th General Assembly, pp. 553-565.
- Lynch, T.J. (1985)** *Data Compression - Techniques and Applications*, Wadsworth (Lifetime Learning Publications), California, 345 pages.
- Makarovic, B. (1973)** *"Progressive Sampling for Digital Terrain Models"*, The ITC Journal, No.3, pp.397-416
- Makarovic, B. (1975)** *"Amended Strategy for Progressive Sampling"*, The ITC Journal, No.1, pp.117-128.
- Makarovic, B. (1976)** *"A Digital Terrain Model System"*, The ITC Journal, No.1, pp.57-83.
- Makarovic, B. (1977)** *"Composite Sampling for Digital Terrain Models"*, The ITC Journal, No.3,

pp.406-433.

- Makarovic, B. (1983)** *"A Test on Compression of Digital Terrain Model Data"*, The ITC Journal, No.2, pp.133-138.
- Mark, D.M. (1975)** *"Computer Analysis of Topography: A Comparison of Terrain Storage Methods"*, Geografiska Annaler, Vol.57A, No.3-4, pp.179-188.
- Mark, D.M. (1978a)** *"Concepts of Data Structure for Digital Terrain Models"*, Proc. of the Digital Terrain Models (DTM) Symposium, ASP/ACSM, May 9-11, pp.24-31.
- Mark, D.M. (1978b)** *"Topological Properties of Geographic Surfaces: Applications in Computer Cartography"*, First International Advanced Study Symposium on Topological Data Structures for Geographic Information Systems, Harvard Papers on GIS, (Geoffrey Dutton, Editor), Vol.5, Data Structures: Surficial and Multi-Dimensional, 12 pages.
- Mark, D.M. (1983)** *"Automated Detection of Drainage Networks from Digital Elevation Models"*, Proc. of Auto Carto VI, Vol.2, Oct. 16-21, pp.288-298.
- Mark, D.M. (1986)** *"The Use of Quadrees in Geographic Information Systems and Spatial Data Handling"*, Proc. of Auto Carto London, Vol.1, Sep. 14-19, pp.517-526.
- Mark, D.M., Lauzon, J.P. & Cebrian, J.A. (1989)** *"A Review of Quadtree-Based Strategies for Interfacing Coverage Data With Digital Elevation Models in Grid Form"*, International Journal of GIS, Vol.3, No.1, pp.3-14.
- Martin, J.J. (1982)** *"Organization of Geographical Data with Quad Trees and Least Squares Approximation"*, Proc. of IEEE Conference on Pattern Recognition and Image Processing, Las Vegas, Nevada, pp.458-463.
- Matthews, P.A. (1965)** *Radio Wave Propagation VHF and Above*, Chapman and Hall, London, England, 155 pages.
- McCullagh, M.J. (1973)** *"Trend Surface Analysis"*, Computer Applications in the Natural and Social Sciences", Vol.15, pp.2-47.
- McCullagh, M.J. (1979)** *"Triangular Systems in Surface Representation"*, Proc. of Auto Carto IV, Vol.1, Nov. 4-8, pp.146-153.
- McCullagh, M.J. & Ross, C.G. (1980)** *"Delaunay Triangulation of a Random Data Set for Isarithmic Mapping"*, The Cartographic Journal, Vol.17, No.2, Dec., pp.93-99.
- McCullagh, M.J. (1981)** *"Creation of Smooth Contours Over Irregularly Distributed Data Using Local Surface Patches"*, Geographical Analysis, Vol.13, No.1, Jan., pp.51-63.
- McCullagh, M.J. (1983)** *"Transformation of Contour Strings to a Rectangular Grid Based Digital Elevation Model"*, Proc. of Euro-Carto II, Reprint, 18 pages
- McCullagh, M.J. (1987)** *"Digital Terrain Modelling and Visualisation"*, Proc. of Terrain Modelling in Surveying and Civil Engineering, Univ. of Surrey, Apr. 7-9, Univ. of Glasgow, Sep. 1-3, 27 pages.
- McCullagh, M.J. (1988)** *"Terrain and Surface Modelling Systems: Theory and Practice"*, Photogrammetric Record, Vol.72, No.12, Oct., pp.747-779.
- McKenna, D.G. (1987)** *"The Inward Spiral Method: An Improved TIN Generation Technique and Data Structure for Land Planning Applications"*, Proc. of Auto Carto VIII, Mar. 29- Apr. 3, pp.670-679.

- McLain, D.H. (1974) *"Drawing Contours from Arbitrary Data Points"*, The Computer Journal, Vol.17, No.4, Apr., pp.318-324.
- McLain, D.H. (1976) *"Two Dimensional Interpolation from Random Data"*, The Computer Journal, Vol.19, No.2, Feb., pp.178-181.
- Meeks, M.L. (1983) *"VHF Propagation over Hilly, Forested Terrain"*, IEEE Trans. on Antennas and Propagation, Vol.AP-31, No.3, May, pp.483-489.
- Merks, E. (1986) *"An Optimal Parallel Algorithm for Triangulating A Set of Points in the Plane"*, International Journal of Parallel Programming, Vol. 15, No.5, pp.399-411.
- Merrill, R.D. (1973) *"Representation of Contours and Regions for Efficient Computer Search"*, Communications of the ACM, Vol.16, No.2, pp.69-82.
- Miller, C.L. & LaFlamme R.A. (1958) *"The Digital Terrain Model - Theory and Application"*, Photogrammetric Engineering and Remote Sensing, Vol.24, No.3, June, pp.433-442.
- Mirante, A. & Weingarten, N. (1982) *"The Radial Sweep Algorithm for Constructing Triangulated Irregular Networks"*, IEEE Computer Graphics and its Applications, Vol.2, No.3, May, pp.11-21.
- Morris, D.G. & Flavin R.W. (1990) *"A Digital Terrain Model for Hydrology"*, Proc. of the 4th Int. Symposium on Spatial Data Handling, Vol.1, Zurich, July 23-27, pp.250-262.
- Neumyvakin, A.Y. & Yakovlev, A.F. (1986) *"Construction of a Digital Terrain Model"*, Mapping Sciences and Remote Sensing, Vol.2, No.3, pp.227-232.
- Oldham, C.H.G. & Sutherland, D.B. (1955) *"Orthogonal Polynomials: Their Use in Estimating the Regional Effect"*, Geophysics, Vol.20, No.2, Apr., pp.295-306.
- Pavlidis, T. (1982) *Algorithms for Graphic and Image Processing*, Rockville MD, Computer Science Press
- Petrie, G. (1987a) *"Photogrammetric Techniques of Data Acquisition for Terrain Modelling"*, Proc. of Terrain Modelling in Surveying and Civil Engineering, Univ. of Surrey, Apr. 7-9, Univ. of Glasgow, Sep. 1-3.
- Petrie, G. (1987b) *"Terrain Data Acquisition and Modelling from Existing Maps"*, Proc. of Terrain Modelling in Surveying and Civil Engineering, Univ. of Surrey, Apr. 7-9, Univ. of Glasgow, Sep. 1-3.
- Petrie, G. (1987c) *"Data Interpolation and Contouring Methods"*, Proc. of Terrain Modelling in Surveying and Civil Engineering, Univ. of Surrey, Apr. 7-9, Univ. of Glasgow, Sep. 1-3.
- Petrie, G. & Kennie, T.J.M. (1987) *"An Introduction to Terrain Modelling: Applications and Terminology"*, Proc. of Terrain Modelling in Surveying and Civil Engineering, Univ. of Surrey, Apr. 7-9, Univ. of Glasgow, Sep. 1-3.
- Peucker, T.K. & Chrisman, N. (1975) *"Cartographic Data Structures"*, The American Cartographer, Vol.2, No.1, pp.55-69.
- Peucker, T.K. & Douglas, D.H. (1975) *"Detection of Surface-Specific Points by Local Parallel Processing of Discrete Terrain Elevation Data"*, Computer Graphics and Image Processing, Vol.4, pp.375-387.
- Peucker, T.K., Fowler, R.J., Little, J.J. & Mark, D.M. (1976) *"Digital Representation of Three-Dimensional Surfaces by Triangulated Irregular Networks"*, Technical Report No. 10, Research Contract N00014-75-C-0886, Project No. NR 389-171, Office of Naval Research,

- Arlington, Virginia, 63 pages.
- Peucker, T.K. (1978)** *"Data Structures for Digital Terrain Models: Discussion and Comparison"*, First International Advanced Study Symposium on Topological Data Structures for Geographic Information Systems, Harvard Papers on GIS, (Geoffrey Dutton, Editor), Vol.5, Data Structures: Surficial and Multi-Dimensional.
- Peucker, T.K., Fowler, R.J., Little, J.J. & Mark, D.M. (1978)** *"The Triangulated Irregular Network"*, Proc. of the Digital Terrain Models (DTM) Symposium, ASP/ACSM, May 9-11, pp.516-540.
- Peucker, T.K. (1979)** *"Digital Terrain Models - An Overview"*, Proc. of Auto Carto IV, Vol.1, Nov. 4-8, pp.97-107.
- Peuquet, D.J. (1981)** *"An Examination of Techniques for Reformatting Digital Cartographic Data / Part One: The Raster to Vector Process"*, Cartographica, Vol.18, No.1, pp.34-48.
- Peuquet, D.J. (1984)** *"A Conceptual Framework and Comparison of Spatial Data Models"*, Cartographica, Vol.21, No.4, pp.66-113.
- Pfaltz, J.L. (1975)** *"Representation of Geographic Surfaces Within a Computer"*, in Display and Analysis of Spatial Data, (NATO Advanced Study Institute, 1973), Editors J.C. Davis & M.J. McCullagh, John Wiley and Sons, London, pp.210-230.
- Pratt, I.J. (1979)** *"Data Acquisition and Interpolation Methods used for Generating Digital Terrain Models"*, Conference on Computer Graphics and Spatial Analysis, Adelaide, Australia, Aug. 13-15, pp.112-116.
- Press, W.H., Flannery, B.P., Teukolsky, S.A. & Vetterling, W.T. (1986)** *"Numerical Recipes - The Art of Scientific Computing"*, Cambridge University Press, Cambridge, 818 pages.
- Reed, H.R. & Russell, C.M. (1966)** *Ultra High Frequency Propagation*, 2nd Edition, Chapman & Hall Ltd, London, 562 pages.
- Reghbati, H.K. (1981)** *"An Overview of Data Compression Techniques"*, IEEE Computer, Vol.14, No.4, Apr., pp.71-75.
- Rhind, D. (1975)** *"A Skeletal Overview of Spatial Interpolation Techniques"*, Computer Applications, Vol.2, pp.293-309.
- Rom, M. & Bergman, S. (1986)** *"A New Technique for Automatic Contouring and Contour Representation from Machine-Readable Spatial Data"*, The Computer Journal, Vol.29, No.5, pp.467-471.
- Samet, H., Rosenfeld, A., Shaffer, C.A. & Webber, R.E. (1983)** *"Quadtree Region Representation in Cartography: Experimental Results"*, IEEE Transactions on Systems, Man and Cybernetics, Vol.SMC-13, No.6, Nov/Dec, pp.1148-1154.
- Samet, H. (1984)** *"The Quadtree and Related Hierarchical Data Structures"*, ACM Computing Surveys, Vol.16, No.2, pp.187-260.
- Samet, H. (1990a)** *The Design and Analysis of Spatial Data Structures*, Addison Wesley, Reading, Mass., USA.
- Samet, H. (1990b)** *Applications of Spatial Data Structures*, Addison Wesley, Reading, Mass., USA.
- Sampson, R.J. (1975)** *"The SURFACE II Graphics System"*, in Display and Analysis of Spatial Data, (NATO Advanced Study Institute, 1973), Editors J.C. Davis & M.J. McCullagh, John

- Wiley and Sons, London , pp.244-266.
- Scarlato, L. (1989) *"A Compact Terrain Model Based on Critical Topographic Features"*, Proc. of Auto Carto IX, Apr. 2-7, pp.146-155.
- Scarlato, L. & Pavlidis, T. (1991) *"Adaptive Hierarchical Triangulation"*, Proc. of Auto Carto 10, Tech. Papers of ACSM/ASPRS, Vol.6, Baltimore, Maryland, Mar.25-28, pp.234-246.
- Schumaker, L.L. (1976) *"Fitting Surfaces to Scattered Data"*, in Approximation III, Proc. of a Symposium on Approximation Theory, Austin, Texas, Editors Lorentz, G.G., Chui, C.K., Schumaker, L.L., Academic Press, Jan. 18-21, pp.203-268.
- Schut, G.H. (1976) *"Review of Interpolation Methods for Digital Terrain Models"*, Canadian Surveyor, Vol.30, No.5, Dec., pp.389-412.
- Shaffer, C.A. (1989) *"A Full Resolution Elevation Representation Requiring Three Bits per Pixel"*, Proc. of Symposium on Large Spatial Databases (SSD'89), University of California, Santa Barbara, Springer-Verlag Lecture Notes in Computer Science, pp.45-63.
- Shepard, D. (1969) *"A Two-Dimensional Interpolation Function for Computer Mapping of Irregularly Spaced Data"*, Harvard Papers in Theoretical Geography - 'Geography and the Properties of Surfaces' Series, Paper 15, Harvard University, Cambridge, Mass., USA, (Office of Naval Research), 20 pages.
- Shmutter, B. & Doytsher, Y. (1978) *"DTM in the Form of Quadrilaterals and its use for Contour Drawing"*, Proc. of the Digital Terrain Models (DTM) Symposium, ASP/ACSM, May 9-11, pp.255-268.
- Sibson, R. (1978) *"Locally Equiangular Triangulations"*, The Computer Journal, Vol.21, No.3, Mar., pp.243-245.
- Sieber, R. (1986) *"Perception of 3-D Representations"*, in Chapter 5 of A Selected Bibliography on Spatial Data Handling: Data Structures, Generalization and Three-Dimensional Mapping (Editors: R.Sieber & K.Brassel), Geoprocessing Series, Vol.6, Dept. of Geography, Univ. of Zurich, pp.59-76.
- Skidmore, A.K. (1990) *"Terrain Position as Mapped from a Gridded Digital Elevation Model"*, International Journal of GIS, Vol.4, No.1, pp.33-49.
- Sotomayor, D.L.G. (1978) *"Tessellation of Triangles of Variable Precision as an Economical Representation for DTMs"*, Proc. of the Digital Terrain Models (DTM) Symposium, ASP/ACSM, May 9-11, pp.506-515.
- Sowton, M. (1989) *"Digital Data: The Future for Ordnance Survey"*, Proc. of Auto Carto IX, Apr. 2-7, pp.493-504.
- Symmons, A.H. (1982) *"Project ACORN (Computerised Radio Path Prediction)"*, J.R. Signals Inst. (GB), Vol.15, No.4, pp.176-179.
- Tesser, H. & De Mund, E. (1989) *"Adaptive Decomposition of Terrain"*, Unpublished paper presented at ACSM/ASPRS/Auto Carto IX, Baltimore, Apr.2-7, 8 pages.
- Theodossiou, E.I. & Dowman, I.J. (1990) *"Height Accuracy of SPOT"*, Photogrammetric Engineering & Remote Sensing, Vol.56, No.12, Dec., pp.1643-1649.
- Tipper, J.C. (1979) *Surface Modelling Techniques*, Series on Spatial Analysis, No.4, Kansas Geological Survey, University of Kansas, 108 pages.
- United States Geological Survey (1990) *Digital Elevation Models, Data Users Guide 5*, U.S.

- Department of the Interior, Reston, Virginia, 2nd Edition, 51 pages.
- Unwin, D. (1981), *Introductory Spatial Analysis*, Methuen & Co. Ltd., London.
- Van Roessel, J.W. (1988) "*Conversion of Cartesian Coordinates From and To Generalised Balanced Ternary Addresses*", *Photogrammetric Engineering and Remote Sensing*, Vol.54, No.11, Nov., pp.1565-1570.
- Verts, W.T. (1991) "*An $O(n^4)$ Delaunay Triangulation Algorithm: Sometimes it's OK to be Slow and Dirty*", Unpublished Paper Presented at ACSM/ASPRS Annual Convention, Baltimore, Maryland, 10 pages.
- Ware, J.A. & Kidner, D.B. (1991) "*Parallel Implementation of the Delaunay Triangulation Within a Transputer Environment*", *Proceedings of 2nd European Conference on GIS*, Brussels, Apr. 2-5, Vol.2, pp.1199-1208.
- Watson, D.F. (1981) "*Computing the N-Dimensional Delaunay Tessellation with Application to Voronoi Polytopes*", *The Computer Journal*, Vol.24, No.2, Feb., pp.167-172.
- Watson, D.F. (1982) "*ACORD: Automatic Contouring of Raw Data*", *Computers and Geosciences*, Vol.8, No.1, Jan., pp.97-101.
- Weibel, R., Heller, M., Herzog, A. & Brassel, K.E. (1989) "*Approaches to Digital Surface Modeling*", in *Contributions to Digital Terrain Modeling and Display* (Ed. R. Weibel), Geo-Processing Series, Vol.12, Dept. of Geography, University of Zurich, Switzerland, pp.1-23. (Revised version of paper in Proc. of 1st Latin American Conf. in Computers and Geography, San José, Costa Rica, 1989).
- Weibel, R. & DeLotto, J.S. (1989) "*Automated Terrain Classification for GIS Modeling*", in *Contributions to Digital Terrain Modeling and Display* (Ed. R. Weibel), Geo-Processing Series, Vol.12, Dept. of Geography, University of Zurich, Switzerland, pp.25-50. (Revised version of paper in Proc. of GIS/LIS '88, San Antonio, Texas, 1988).
- Weibel, R. (1989) "*Design and Implementation of a Strategy for Computer-Assisted Terrain Generalisation*", in *Contributions to Digital Terrain Modeling and Display* (Ed. R. Weibel), Geo-Processing Series, Vol.12, Dept. of Geography, University of Zurich, Switzerland, pp.51-65. (Paper presented at 14th Conference of the ICA, Budapest, Hungary, 1989).
- Weibel, R. & Herzog, A. (1989) "*Automated Construction of Panoramic Views from Digital Terrain Models*", in *Contributions to Digital Terrain Modeling and Display* (Ed. R. Weibel), Geo-Processing Series, Vol.12, Dept. of Geography, University of Zurich, Switzerland, pp.67-98. (Translation of German original).
- Weibel, R. & Heller, M. (1990) "*A Framework for Digital Terrain Modelling*", *Proc. of the 4th Int. Conference on Spatial Data Handling*, Vol.1, Zurich, July 23-27, pp.219-229.
- Welch, T.A. (1984) "*A Technique for High-Performance Data Compression*", *IEEE Computer*, Vol.17, No.6, June, pp.8-19.
- Whitten, E.H.T. (1970) "*Orthogonal Polynomial Trend Surfaces for Irregularly Spaced Data*", *Journ. of the Int. Association for Mathematical Geology*, Vol.2, No.2, pp.141-152.
- Whitten, E.H.T. (1975) "*The Practical Use of Trend-Surface Analyses in the Geological Sciences*", in *Display and Analysis of Spatial Data*, (NATO Advanced Study Institute, 1973), Editors J.C. Davis & M.J. McCullagh, John Wiley and Sons, London, pp.282-297.
- Williams, C.M. (1986) "*The Geometric Modeling and Compression of Terrain Data*", *Proc. of 2nd Int. Conference on Spatial Data Handling*, Seattle, July 5-10, pp.158-170.

- Wolf, G.W. (1991)** *"Characterization of Functions Representing Topographic Surfaces"*, Proc. of Auto-Carto 10, Tech. Papers of ACSM/ASPRS, Vol.6, Baltimore Maryland, Mar.25-28, pp.186-204.
- Woo, T.C. (1985)** *"A Combinatorial Analysis of Boundary Data Structure Schemata"*, IEEE Computer Graphics & Applications, Vol.5, No.3, pp.19-27.
- Yoeli, P. (1975)** *"Compilation of Data for Computer-Assisted Relief Cartography"*, in Display and Analysis of Spatial Data, (NATO Advanced Study Institute, 1973), Editors J.C. Davis & M.J. McCullagh, John Wiley and Sons, London, pp.352-367.
- Yoeli, P. (1983a)** *"Digital Terrain Models and their Cartographic and Cartometric Utilisation"*, The Cartographic Journal, Vol.20, No.1, June, pp.17-22.
- Yoeli, P. (1983b)** *"About Cartographic Contouring with Computers"*, Proc. of Auto Carto VI, Vol.2, October 16th-21st, pp.262-266.
- Yoeli, P. (1986)** *"Computer Executed Production of a Regular Grid of Height Profiles from Digital Contours"*, The American Cartographer, Vol.13, No.3, pp.219-229.

Appendix A

Digital Elevation Model Data

Digital contour and elevation data of Great Britain are now being made available through the Ordnance Survey (O.S.). The data has been produced from the contours of the O.S. 1:50,000 scale Landranger Map Series. Data is supplied in 20 x 20 kilometre (km) cells, the southwest corner of which is an even-numbered 10 km National Grid (NG) value. Data for southern Britain is now available, with full coverage expected to be complete by 1992. The terrain data is represented as height values at each intersection of a 50 metre horizontal grid, so there are 401 x 401 or 160,801 grid node elevations in each cell. This data has been mathematically interpolated from the contour data described above, and the heights have been rounded to a vertical resolution of one metre. A description of the inherent errors within these data sets and the research being undertaken to correct them is described by Morris & Flavin (1990).

The accuracy of this data has been sample tested by O.S. for some of the data sets produced, with variations in the accuracy depending on the nature of the terrain. These results range from a RMS error of 1.7 metres in hilly rural areas to 2.5 metres in an urban lowland area. Whilst these figures seem acceptable, there is no indication of largest absolute error. However, it has been shown in this research (for example, Table 5.21 in Chapter Five), that a regular grid DEM with similar RMS errors may be expected to cause up to 5% of terrain profiles to be in corresponding radio path loss error by more than ± 6 dBs. It can be assumed therefore, that some of these data sets may create a similar number of path loss errors. A further criticism of the available data is the lack of detailed information concerning buildings, forests and other man-made obstructions.

Throughout this research, two neighbouring O.S. data sets for an area of South Wales are used as test data for the derivation of all the DEMs. These are referenced as ST06 and ST08 in O.S. grid reference cell ST (Figure A1.1).

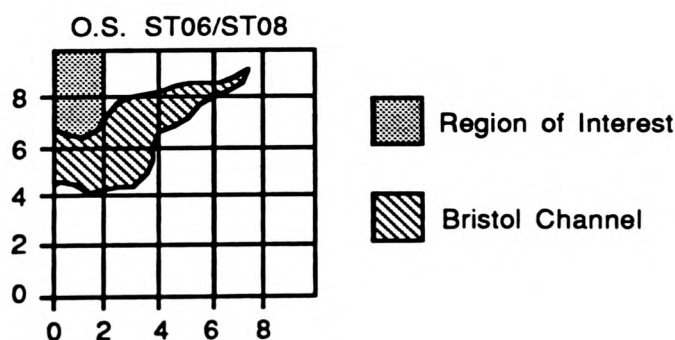


Figure A1.1 - O.S. Grid Cell ST Referencing 20 x 20 km Data Sets ST06 and ST08.

The first surface ST06 (lower left grid position at O.S. Coordinates [300,000m, 160,000m] or approximately longitude 3° 24' W, latitude 51° 13' N) represents a coastal plain flanking the Bristol Channel and includes the ports of Cardiff and Barry. Over 35% of this terrain model is at sea level. There is a narrow coastal plain, cliffed bays with in general, relatively small relief. The ST08 DEM (at [300,000m, 180,000m] or longitude 3° 26' W, latitude 51° 24' N) is typical of much of the terrain in the South Wales valleys (ie. hilly or mountainous regions of variable relief, cut by steep, deep valleys). Its main features are the Taff and Rhondda Valleys intersecting at Pontypridd and flowing towards Cardiff in the southeast. The Pennant sandstone forms the high ground in the region, which is Carboniferous in age. The drainage cuts across the east-west dipping lithology, therefore the northwest-southeast drainage pattern has been superimposed.

These two data sets represent differing types of terrain, and as such are useful in illustrating the advantages and disadvantages of the surface modelling methods, especially with regard to surface morphology. Contour maps and isometric projections of these data sets are illustrated overleaf. The surface statistics for these elevation models are shown below in Table A1.1.

Surface Statistic	ST06	ST08
Elevation Range	0 - 135 m	19 - 470 m
Average Elevation	31.32 m	173.17 m
Standard Deviation	34.39 m	88.43 m

Table A1.1 - ST06/08 Surface Statistics

The different nature of these two terrain models can be more clearly illustrated with the cumulative frequency distribution of elevations (Figure A1.2), which shows the greater relief of ST08.

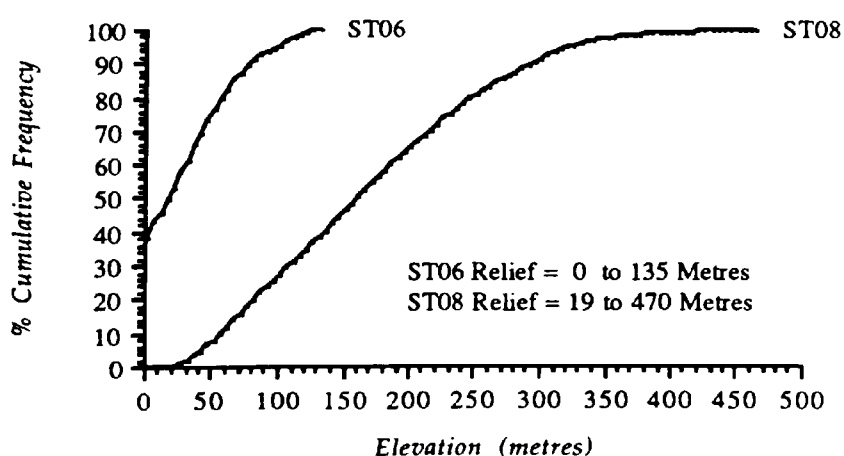
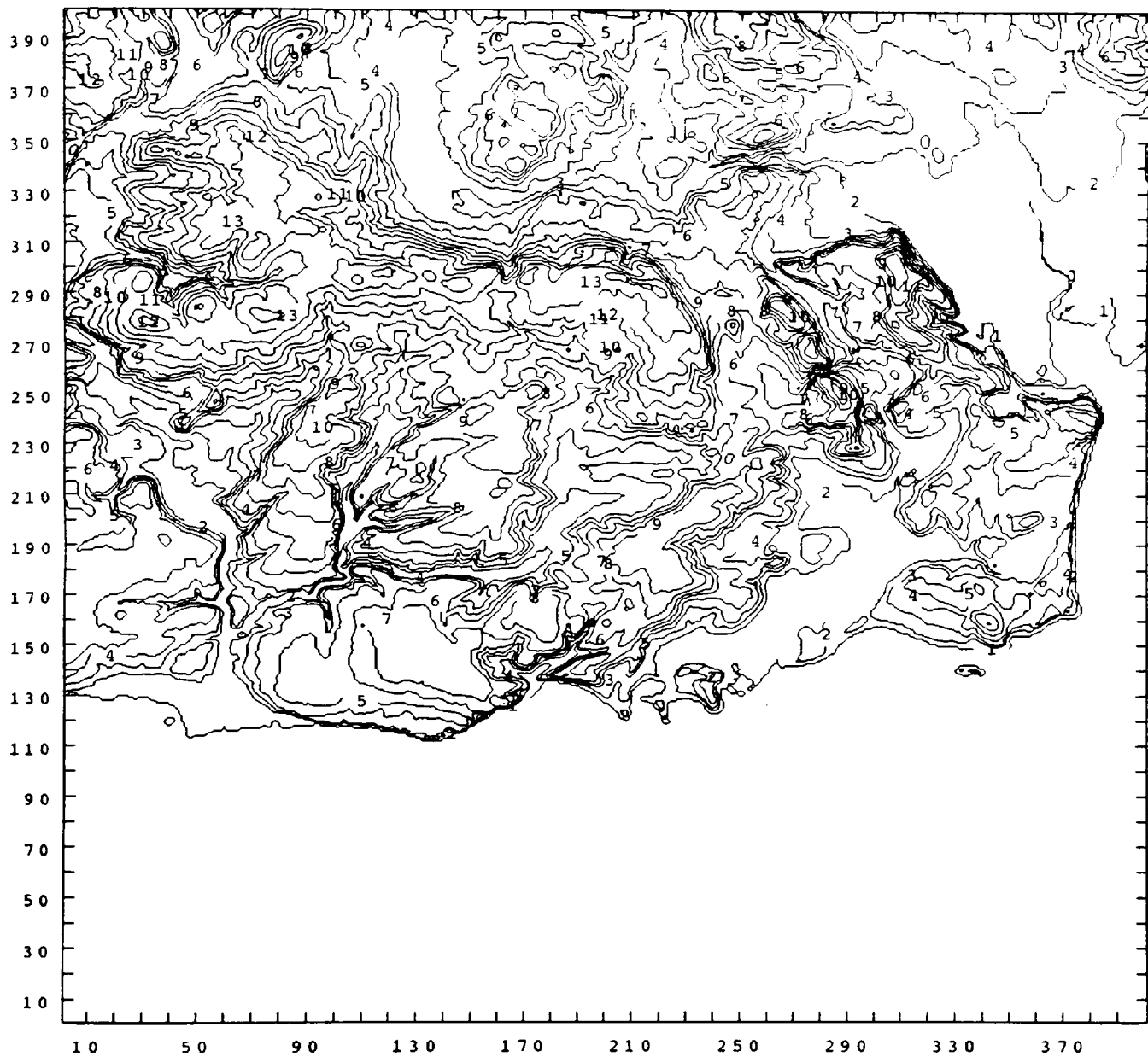


Figure A1.2 - Cumulative Frequency of Elevations for ST06 and ST08.

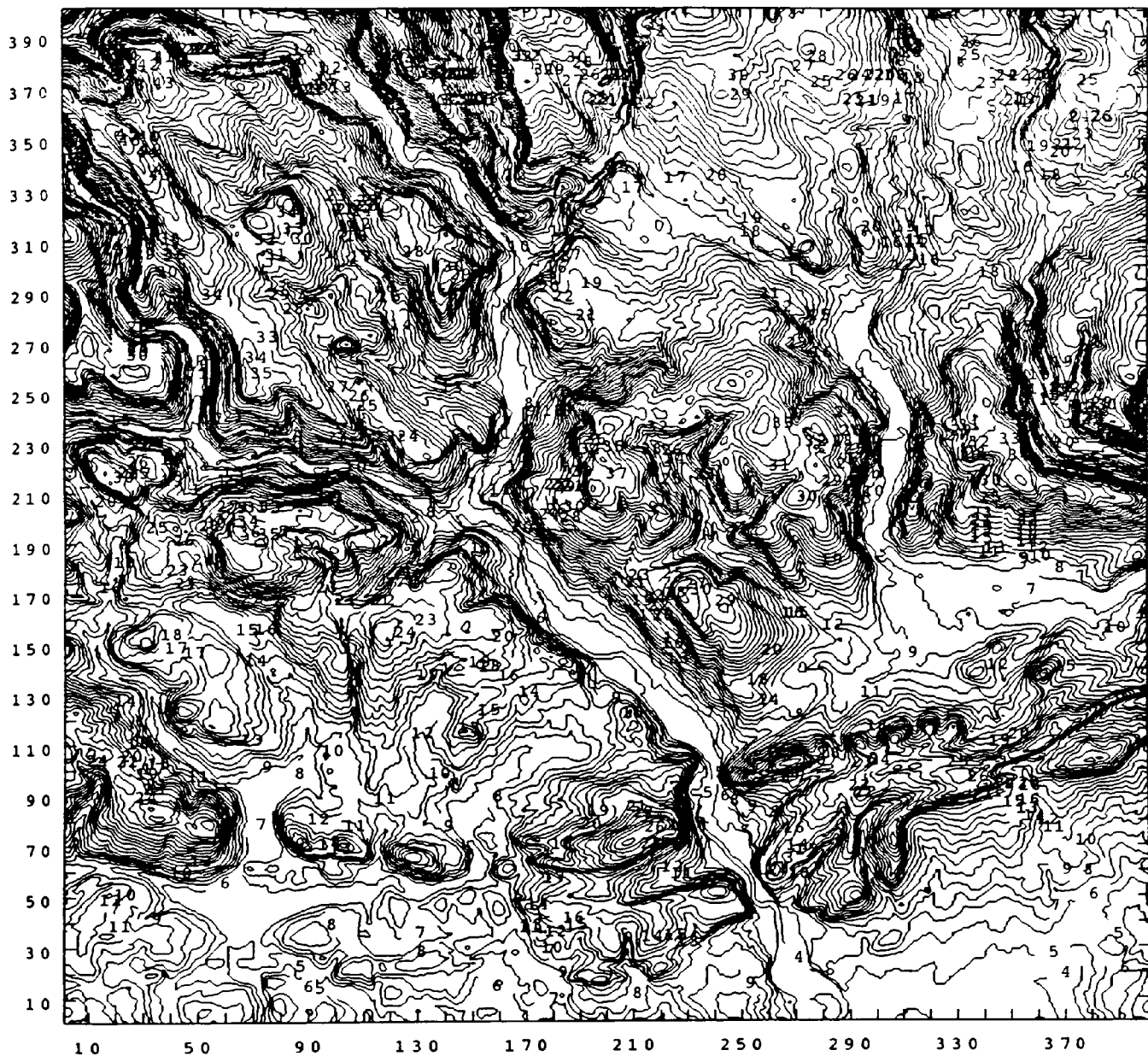
ST06 - 20 km x 20 km - 401 x 401 Grid Nodes Sampled at 50 metre Intervals



Contour Key (Elevations in Metres)					
1	0	6	50	11	100
2	10	7	60	12	110
3	20	8	70	13	120
4	30	9	80	14	130
5	40	10	90	15	140

Figure A1.3 - ST06.

ST08 - 20 km x 20 km - 401 x 401 Grid Nodes Sampled at 50 metre Intervals



Contour Key (Elevations in Metres)									
1	0	11	100	21	200	31	300	41	400
2	10	12	110	22	210	32	310	42	410
3	20	13	120	23	220	33	320	43	420
4	30	14	130	24	230	34	330	44	430
5	40	15	140	25	240	35	340	45	440
6	50	16	150	26	250	36	350	46	450
7	60	17	160	27	260	37	360	47	460
8	70	18	170	28	270	38	370	48	470
9	80	19	180	29	280	39	380		
10	90	20	190	30	290	40	390		

Figure A1.4 - ST08.

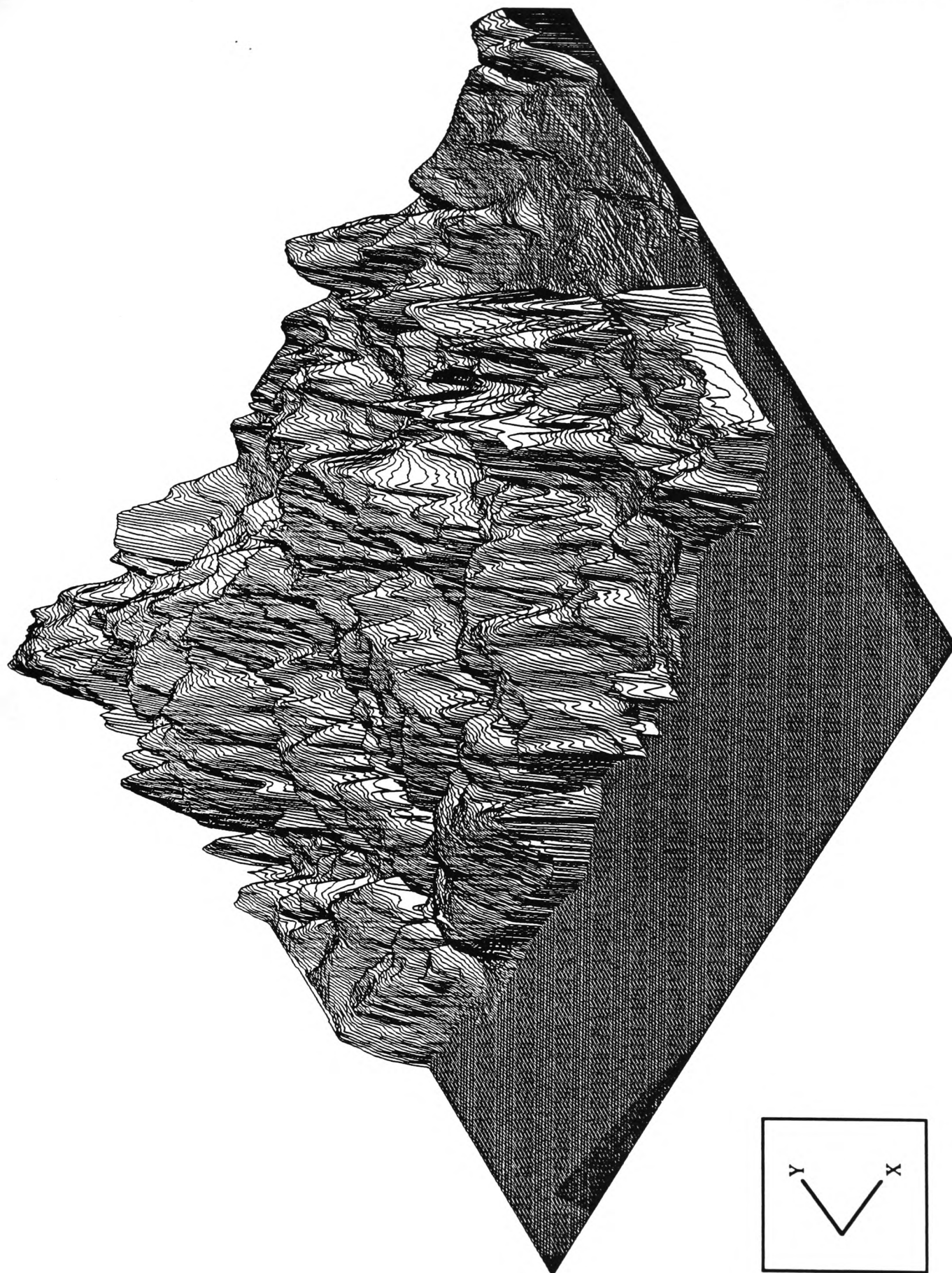


Figure A1.5 - Isometric Projection of ST06. (Vertical Scale = 0 to 135 Metres).

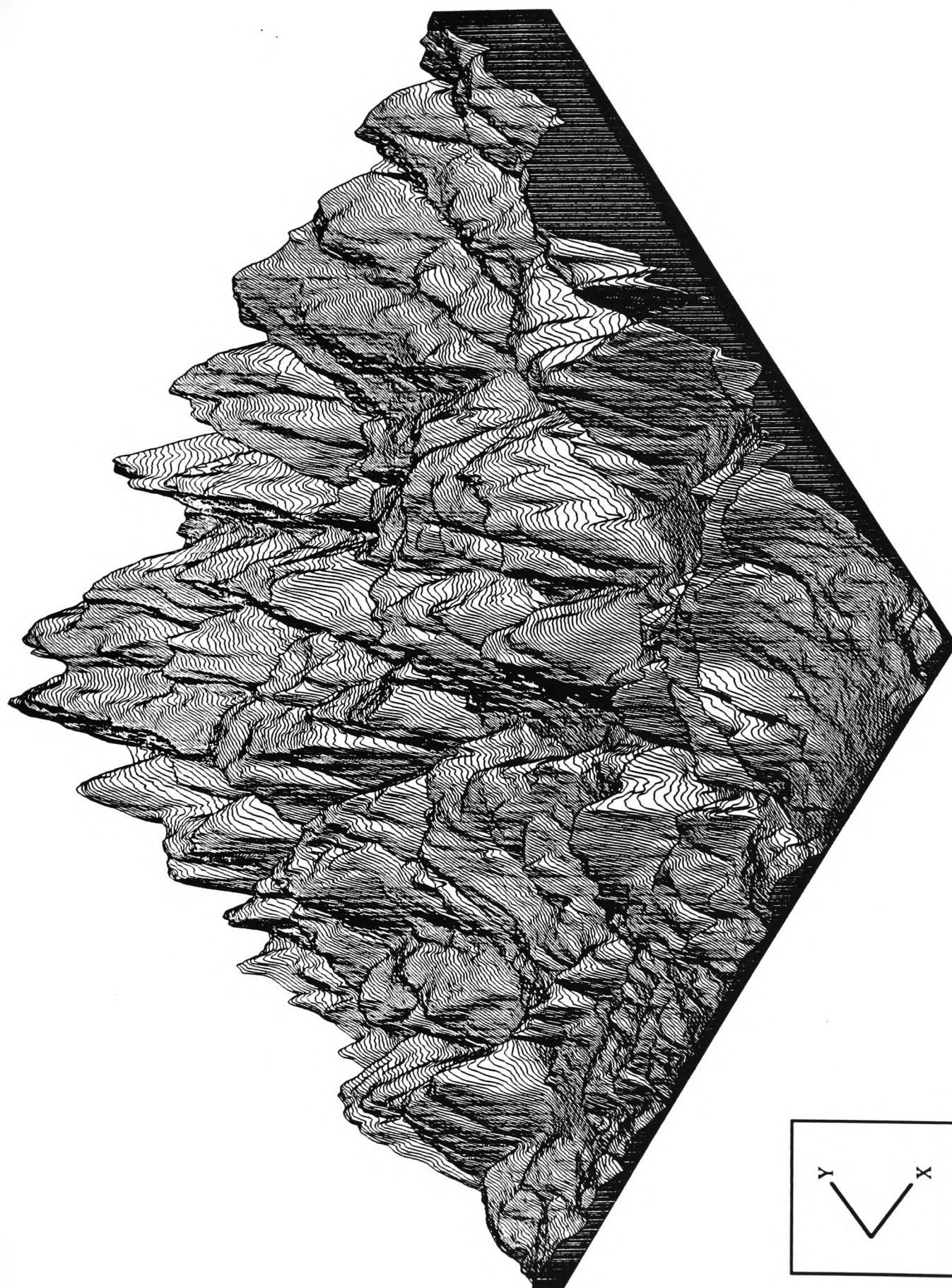


Figure A1.6 - Isometric Projection of ST08. (Vertical Scale = 19 to 470 Metres).

Graphical Routines

Contour Maps

The algorithm for producing contour maps is described by Heap and Pink (1969), and implemented in the Numerical Algorithms Group (NAG) software package. With a given contour level, the algorithm first locates segments between adjacent grid points on the boundary of the region and along horizontal lines which the contour must cross. All open contours are then traced from the boundary through each elementary grid rectangle until the boundary is reached again. Closed contours are then detected and similarly traced by looking for crossed segments through which no contour of the desired level has yet been traced. For this contour tracing routine to work satisfactorily, it is essential that the grid spacing is sufficiently fine for linear interpolation to be adequate, and for it to be assumed that a contour will not cross a grid line more than once between adjacent grid points. Once the contours have been traced, the points are joined together by straight line segments. The use of smoothed curves joining the points results in contours 'colliding' or crossing each other, especially for sparse grids. The use of large data sets, such as the 401x401 O.S. data sets, will make the contours appear smooth.

Isometric Maps

An orthogonal projection is a projection from infinity onto a plane which is at right angles to the line of sight, whilst an isometric projection is a special case of an orthogonal projection in which the coordinate axes make equal angles with the line of sight. The algorithm used to draw the isometric projections is based on Hall (1977), and has been implemented by NAG. The surface is drawn in sections parallel to the x-axis and then parallel to the y-axis. Each section is scanned in turn, starting with the section nearest the viewpoint (selected by the user). The first section is drawn in full as all points are visible, and the points are stored as the visible profile. Successive sections are then scanned and the visibility of points tested against the visible profile. When adjacent points change from visible to hidden or hidden to visible, the point where the line disappears or appears is calculated using linear interpolation. The visible part of the section is drawn using straight line segments and the visible profile is updated to contain the maximum value on each vertical. (The user can select the viewing point).

Appendix B

Radio Path Loss Prediction: Algorithm and Examples

Jones & Knight's original study (1987) aimed to identify the sensitivity of path loss prediction with respect to elevation errors in the ground profiles. The formulae in their algorithm are derived from a number of existing algorithms, including those of Edwards & Durkin (1969); Dadson (1979); Meeks (1983) and others. A review of a number of prediction methods, including these, applied to digital data of hilly and mountainous terrain and compared to measured field-strength recordings is presented by Grosskopf (1987). The basic principle of the algorithm used in this study considers the significant diffracting features of the terrain profiles as knife edges. This multiple diffraction algorithm considers up to three edges and is based upon that of Jones & Knight (1987), with some amendments to the identification of knife edges, particularly those on 'rounded hills' (see Chapter One, Section 1.7). A FORTRAN 77 program listing of this implemented algorithm is provided by Jones & Knight (1987), whilst amendments to identify critical profiles are presented in their later report (1988). The specific details of the algorithm will therefore not be reiterated here, but the features and formulae of the algorithm are fairly standard and can be found within the relevant literature.

Given the position of the base station or transmitter (T) and the receiver (R) at a horizontal distance (d) apart, a ground profile of terrain heights H_i , ($i=1,2,\dots,n$) at horizontal distances d_i from T is obtained. The distances are equally spaced at an interval of δ , usually 50 metres. The heights H_i are first corrected for earth curvature (this has very little effect on relatively short paths). The radio path loss is then taken to be the sum of a number of components:

- (a) Clutter Loss and Polarisation Loss are not considered and are taken to be zero.
- (b) Free Space Loss, calculated for d at the frequency (f).
- (c) Reflection Loss
- (d) Siting Loss (calculated if the Reflection Loss is zero), due to inadequate clearance near the transmitter or receiver.
- (e) Diffraction Loss, calculated for up to three diffraction edges.
- (f) Other Losses, depending upon the frequency and whether the path is obstructed.

This algorithm is illustrated for five profiles through the original regular grid DEM of ST08 (Figure B.1), representing examples of 0, 1, 2, 3 and > 3 diffraction edges (Figures B.2 to B.6). The total radio path loss for each of these profiles is calculated at frequencies of 200 to 1800 MHz, together with a breakdown of the individual loss components.

Examples: (1) Point-to-Point Losses

Consider the five following radio path profiles for ST08 (Figure B.1, Table B.1):

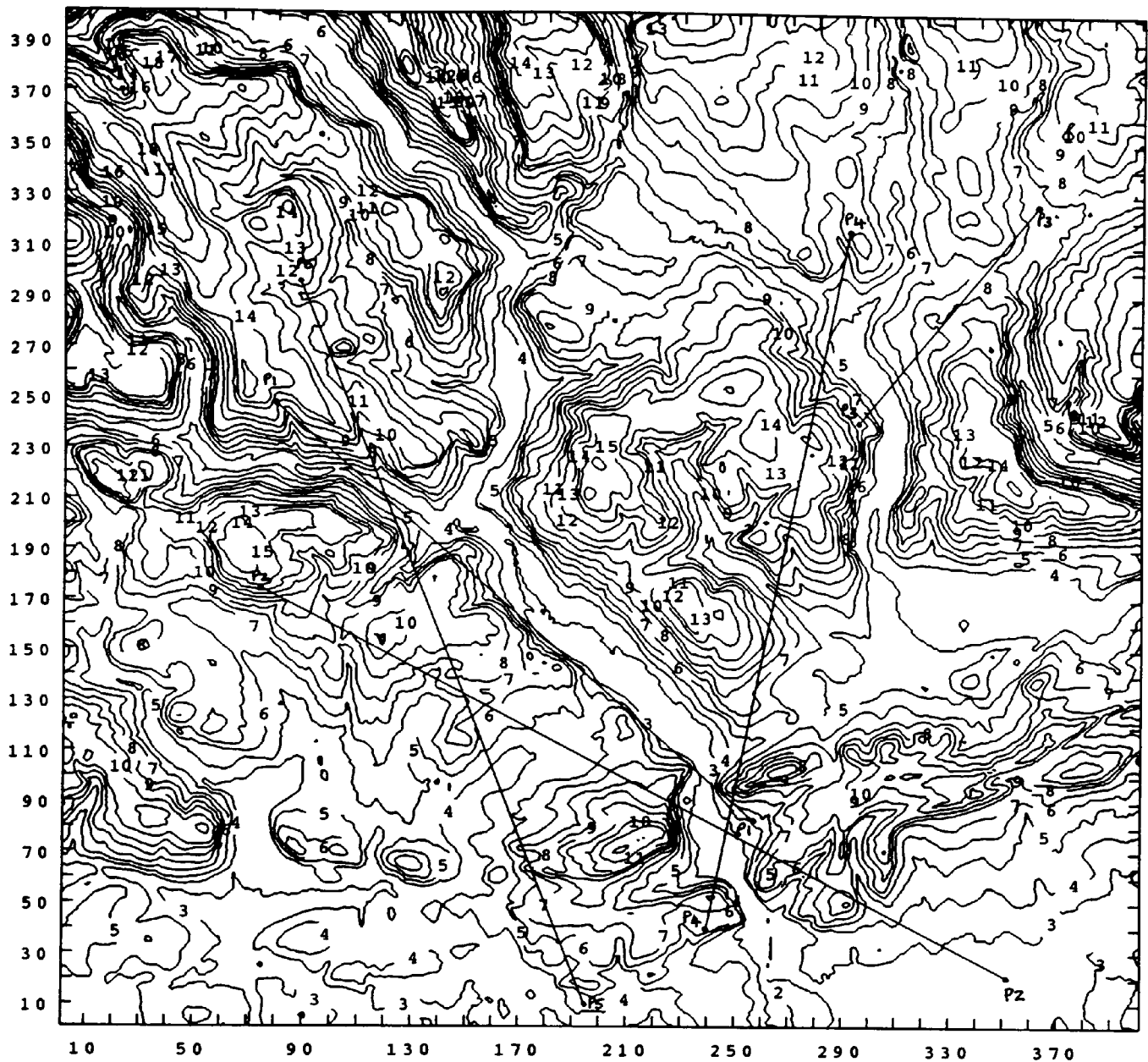


Figure B.1 - Five Point-to-Point Radio Path Profiles for ST08.

Profile	Obstructions	Path Length	Profile Grid Coordinates			
			X1	Y1	X2	Y2
1	0	12.09 km	256.99	83.13	79.45	247.25
2	1	15.87 km	351.74	20.89	73.87	174.24
3	2	5.40 km	295.68	240.55	361.63	326.11
4	3	14.08 km	291.81	315.84	239.53	39.12
5	> 3	15.24 km	88.09	295.61	194.68	10.11

Table B.1 - Coordinates, Length and Number of Obstructions for Profiles in Figure B.4.

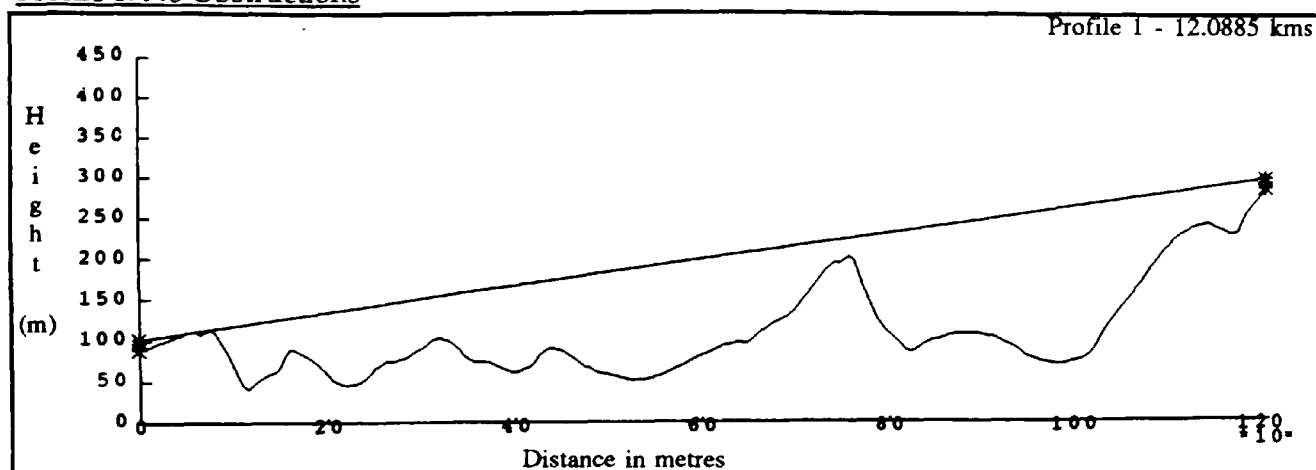
Profile 1: No Obstructions

Figure B.2 - ST08 Point-to-Point Profile: No Obstructions.

Freq. (MHz)	F. Space Loss	Reflect. Loss	Siting Loss T	Siting Loss R	Diffra. Loss ip	Diffra. Loss iq	Diffra. Loss ir	Other Loss	Total Loss
200	100.12	-	6.26	-	-	-	-	-	106.38
400	106.14	-	-	-	-	-	-	-	106.14
600	109.66	-	-	-	-	-	-	3.43	113.10
900	113.18	-	-	-	-	-	-	3.96	117.15
1400	117.02	-	-	-	-	-	-	4.54	121.56
1800	119.20	-	-	-	-	-	-	4.87	124.07

Table B.2 - ST08 Point-to-Point Profile: No Obstructions - Losses at 200MHz to 1800 MHz.

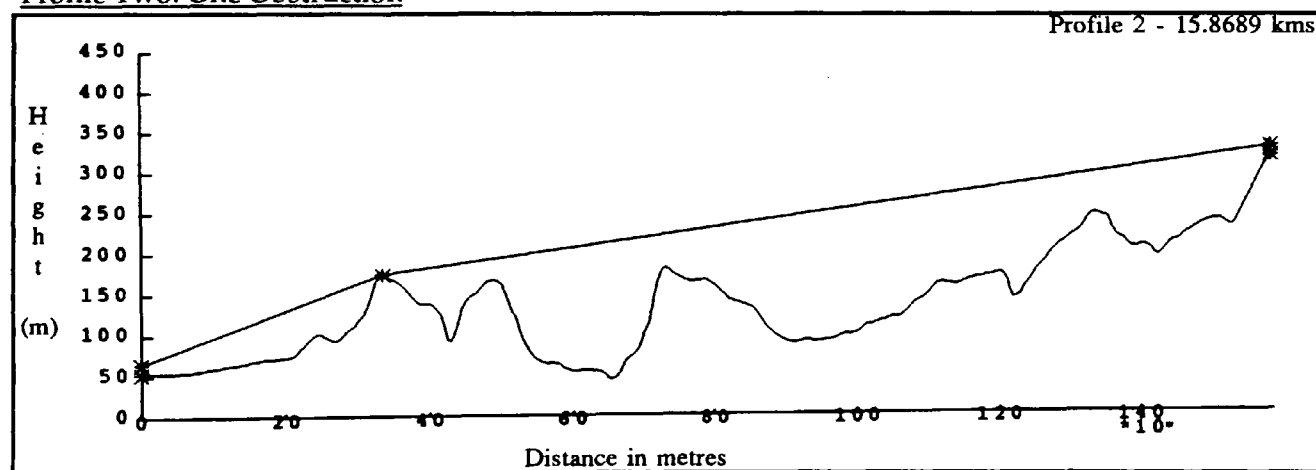
Profile Two: One Obstruction

Figure B.3 - ST08 Point-to-Point Profile: One Obstruction.

Freq. (MHz)	F. Space Loss	Reflect. Loss	Siting Loss T	Siting Loss R	Diffra. Loss ip	Diffra. Loss iq	Diffra. Loss ir	Other Loss	Total Loss
200	102.48	-	1.57	-	15.17	-	-	-	119.23
400	108.50	-	-	-	17.83	-	-	-	126.34
600	112.03	-	-	-	19.40	-	-	1.96	133.38
900	115.55	-	-	-	21.16	-	-	4.69	141.39
1400	119.39	-	-	-	23.08	-	-	7.67	150.12
1800	121.57	-	-	-	24.17	-	-	9.36	155.09

Table B.3 - ST08 Point-to-Point Profile: One Obstruction - Losses at 200MHz to 1800 MHz.

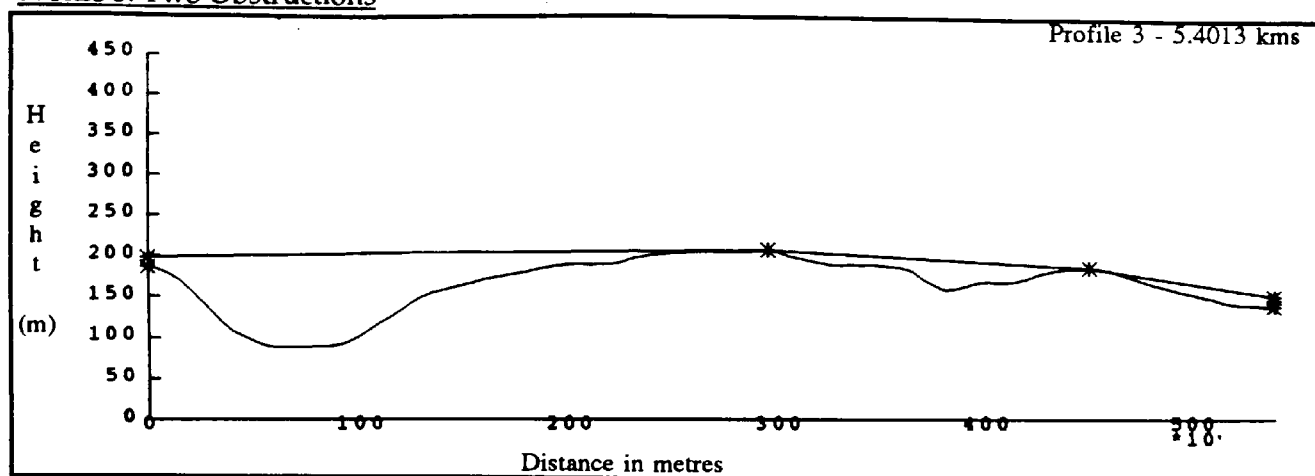
Profile 3: Two Obstructions

Figure B.4 - ST08 Point-to-Point Profile: Two Obstructions.

Freq. (MHz)	F. Space Loss	Reflect. Loss	Siting Loss T	Siting Loss R	Diffra. Loss ip	Diffra. Loss iq	Diffra. Loss ir	Other Loss	Total Loss
200	93.12	10.08	-	-	11.29	11.41	-	-	125.91
400	99.14	4.06	-	-	13.19	13.34	-	-	129.73
600	102.66	-	-	-	14.51	14.68	-	1.96	134.35
900	106.19	-	-	-	15.97	16.16	-	4.69	143.00
1400	110.02	-	-	-	17.69	17.88	-	7.67	153.26
1800	112.21	-	-	-	18.70	18.91	-	9.36	159.17

Table B.4 - ST08 Point-to-Point Profile: Two Obstructions - Losses at 200MHz to 1800 MHz.

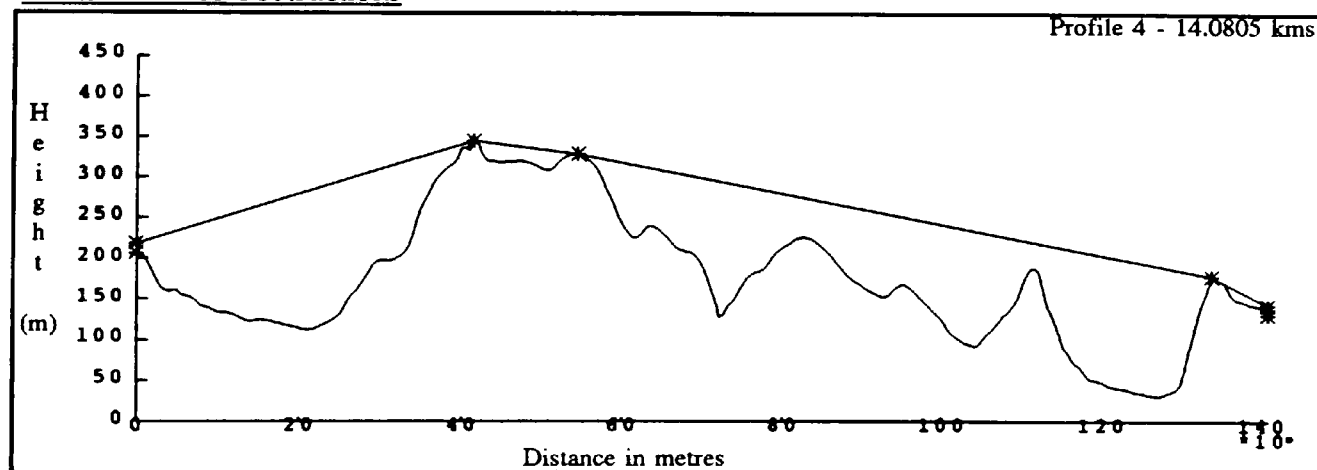
Profile 4: Three Obstructions

Figure B.5 - ST08 Point-to-Point Profile: Three Obstructions.

Freq. (MHz)	F. Space Loss	Reflect. Loss	Siting Loss T	Siting Loss R	Diffra. Loss ip	Diffra. Loss iq	Diffra. Loss ir	Other Loss	Total Loss
200	101.44	-	-	7.07	17.11	13.36	8.23	-	147.21
400	107.46	-	-	1.96	19.85	15.73	9.12	-	154.12
600	110.99	-	-	-	21.62	17.29	9.79	1.96	161.65
900	114.51	-	-	-	23.38	18.93	10.60	4.69	172.10
1400	118.35	-	-	-	25.30	20.72	11.63	7.67	183.66
1800	120.53	-	-	-	26.39	21.82	12.30	9.36	190.39

Table B.5 - ST08 Point-to-Point Profile: Three Obstructions - Losses at 200MHz to 1800 MHz.

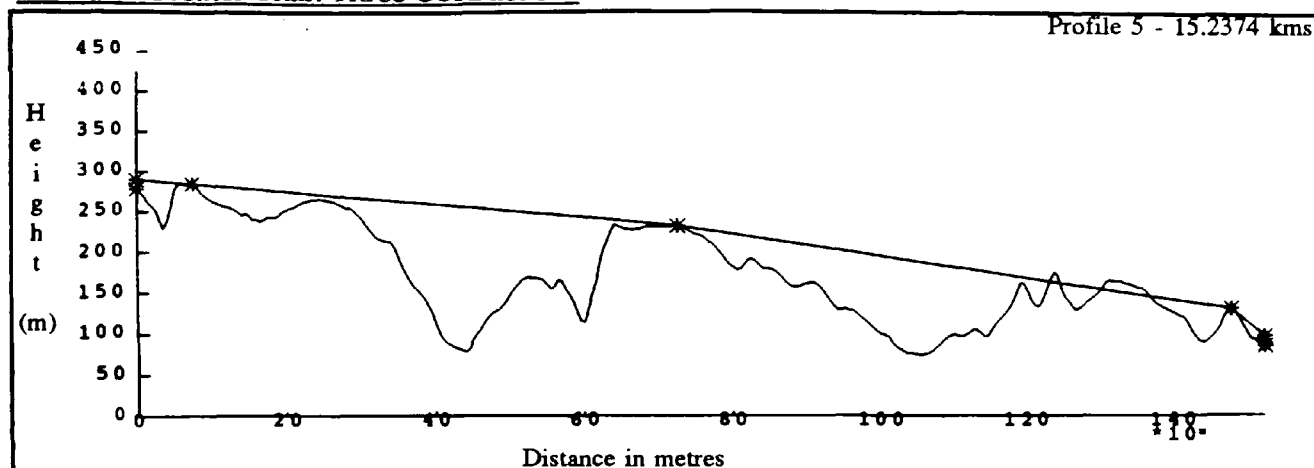
Profile 5: Greater Than Three Obstructions

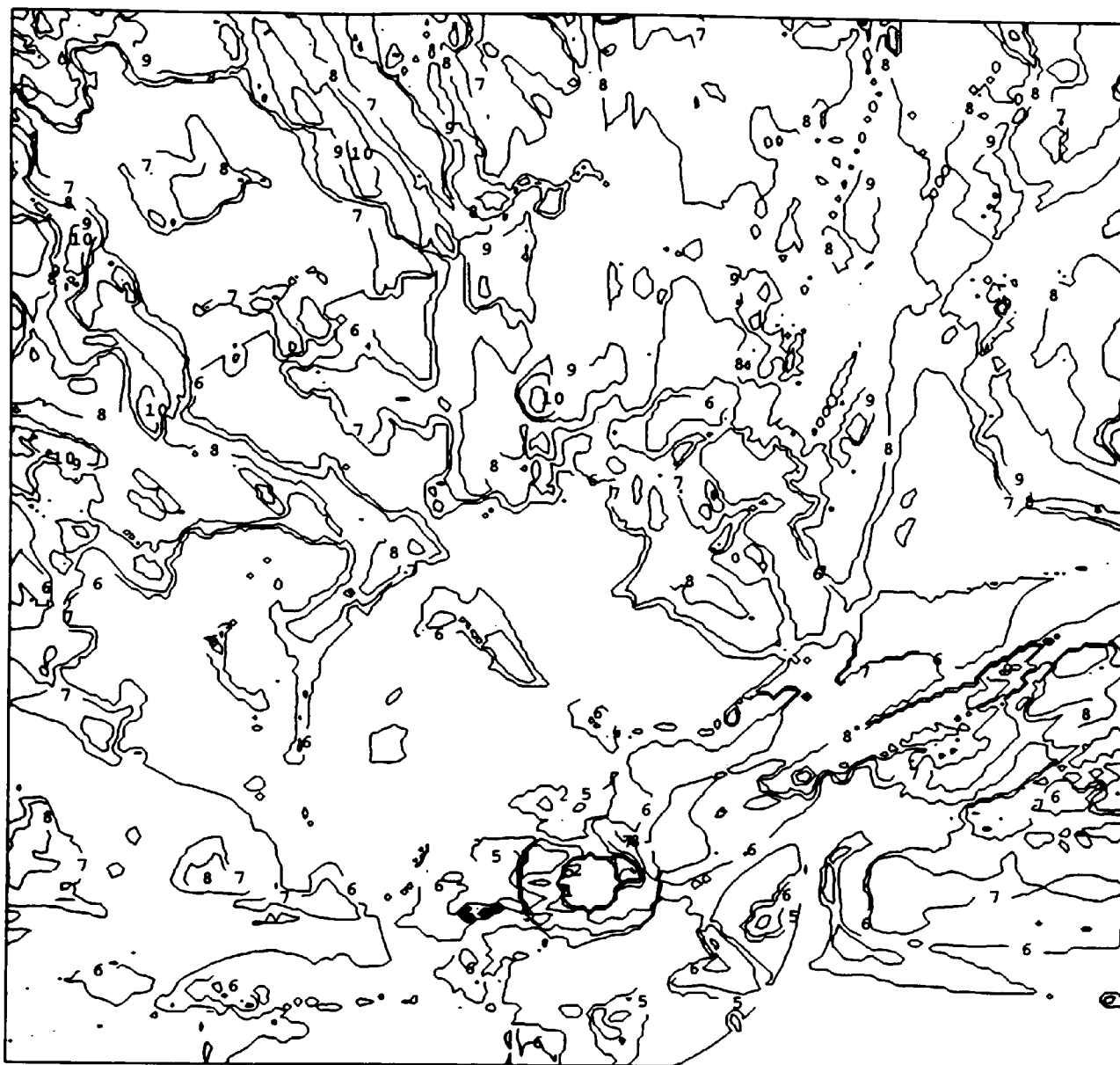
Figure B.6 - ST08 Point-to-Point Profile: > Three Obstructions.

Freq. (MHz)	F. Space Loss	Reflect. Loss	Siting Loss T	Siting Loss R	Diffra. Loss ip	Diffra. Loss iq	Diffra. Loss ir	Other Loss	Total Loss
200	102.13	-	-	2.17	6.09	16.53	9.13	-	136.05
400	108.15	-	-	-	6.12	19.21	10.36	-	143.85
600	111.67	-	-	-	6.15	20.97	11.27	1.96	152.02
900	115.19	-	-	-	6.18	22.73	12.32	4.69	161.12
1400	119.03	-	-	-	6.23	24.65	13.64	7.67	171.21
1800	121.21	-	-	-	6.26	25.74	14.47	9.36	177.04

Table B.6 - ST08 Point-to-Point Profile: >Three Obstructions - Losses at 200MHz to 1800 MHz.

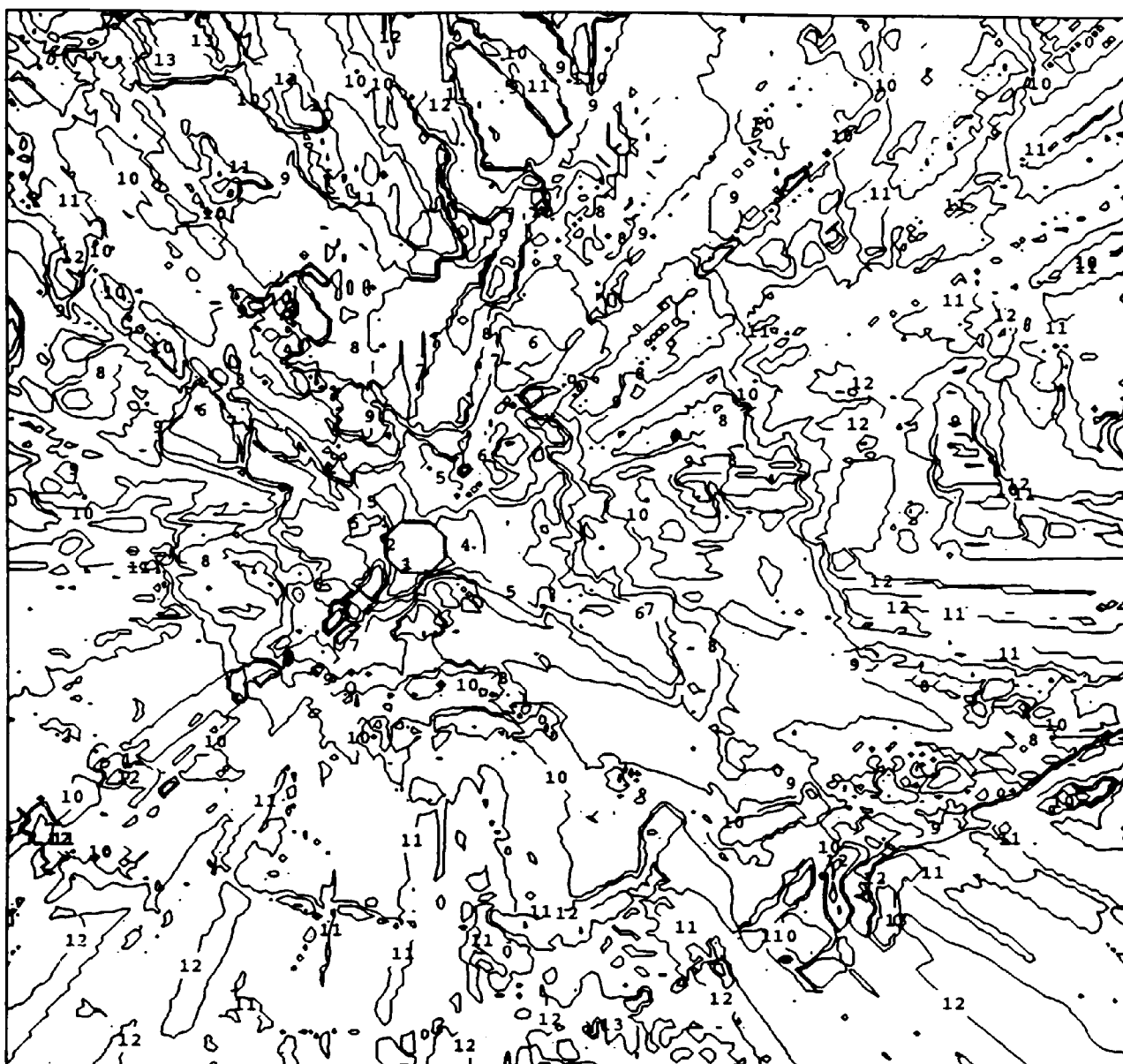
Examples: (2) Field-of-View Losses

Figures B.7 and B.8 represent contour maps of radio path losses at mobile stations within the enclosed area of data set ST08. The transmitter is considered as a fixed-base station, whilst the receivers are considered at variable locations within the terrain. The radio path loss is calculated at each mobile station, such that a matrix of attenuation losses can be formulated. Figure B.7 represents the field losses in decibels with a base station at grid coordinates 209, 71 (peak of Garth Mountain at 302 metres), transmitting at a frequency of 200 MHz. Figure B.8 represents the field losses with a base station at 146, 199 (local minimum of Pontypridd at 49 metres), transmitting at a frequency of 200 MHz. Figures B.9 and B.10 represent the radio field losses at the same transmitter base stations, but at frequencies of 200, 400, 600, 900, 1400 and 1800 MHz.



Contour Key			
1	50 dB	8	120 dB
2	60 dB	9	130 dB
3	70 dB	10	140 dB
4	80 dB	11	150 dB
5	90 dB	12	160 dB
6	100 dB	13	170 dB
7	110 dB	14	180 dB

Figure B.7 - ST08 Field Losses (in dB) For a Base Station at (209,71) Transmitting at 200 MHz.



Contour Key			
1	50 dB	8	120 dB
2	60 dB	9	130 dB
3	70 dB	10	140 dB
4	80 dB	11	150 dB
5	90 dB	12	160 dB
6	100 dB	13	170 dB
7	110 dB	14	180 dB

Figure B.8 - ST08 Field Losses (in dB) For a Base Station at (146,199) Transmitting at 200 MHz.

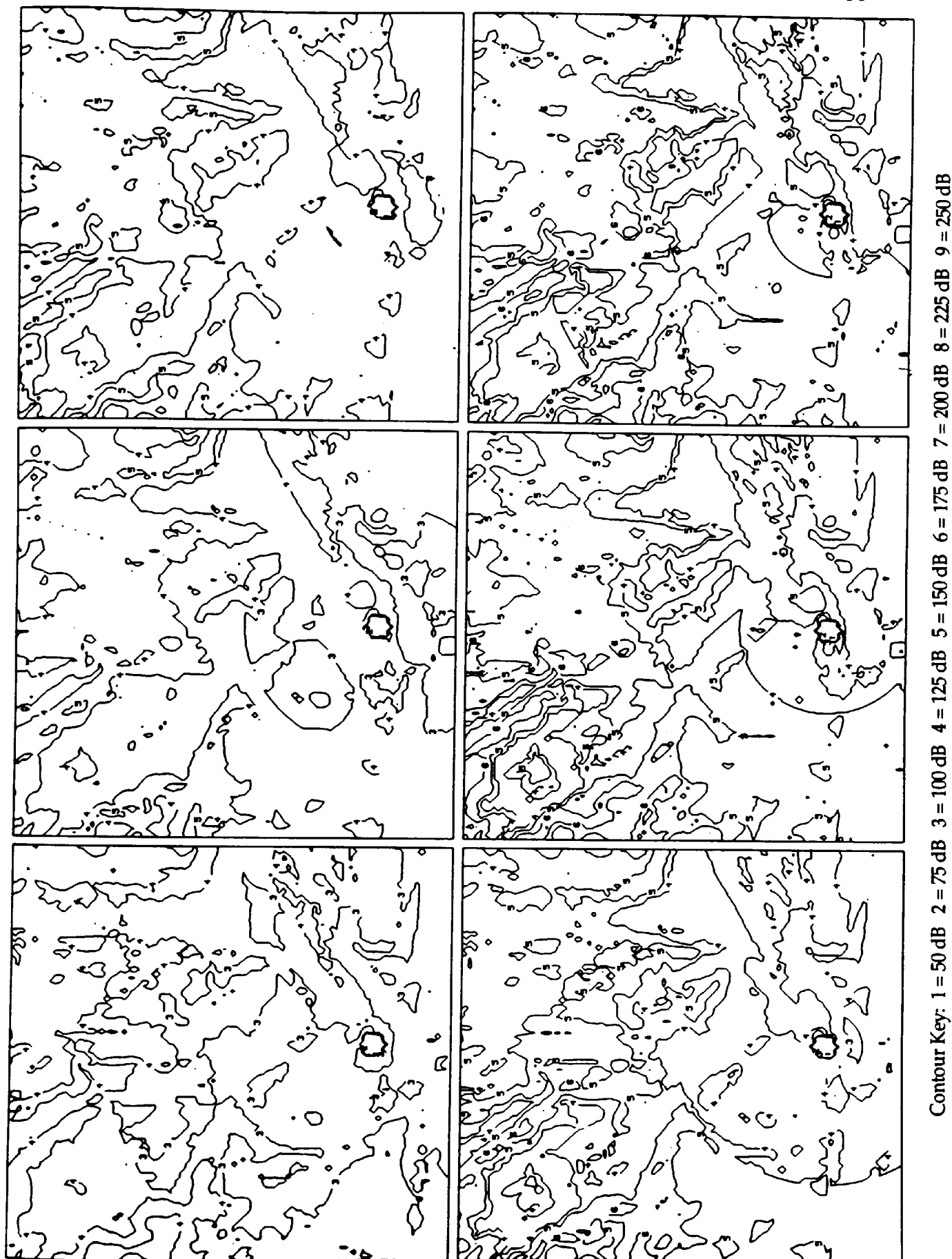


Figure B.9 - ST08 Field Losses (in dB) For a Base Station at (209,71)
Transmitting at 200, 400, 600, 900, 1400 and 1800 MHz.

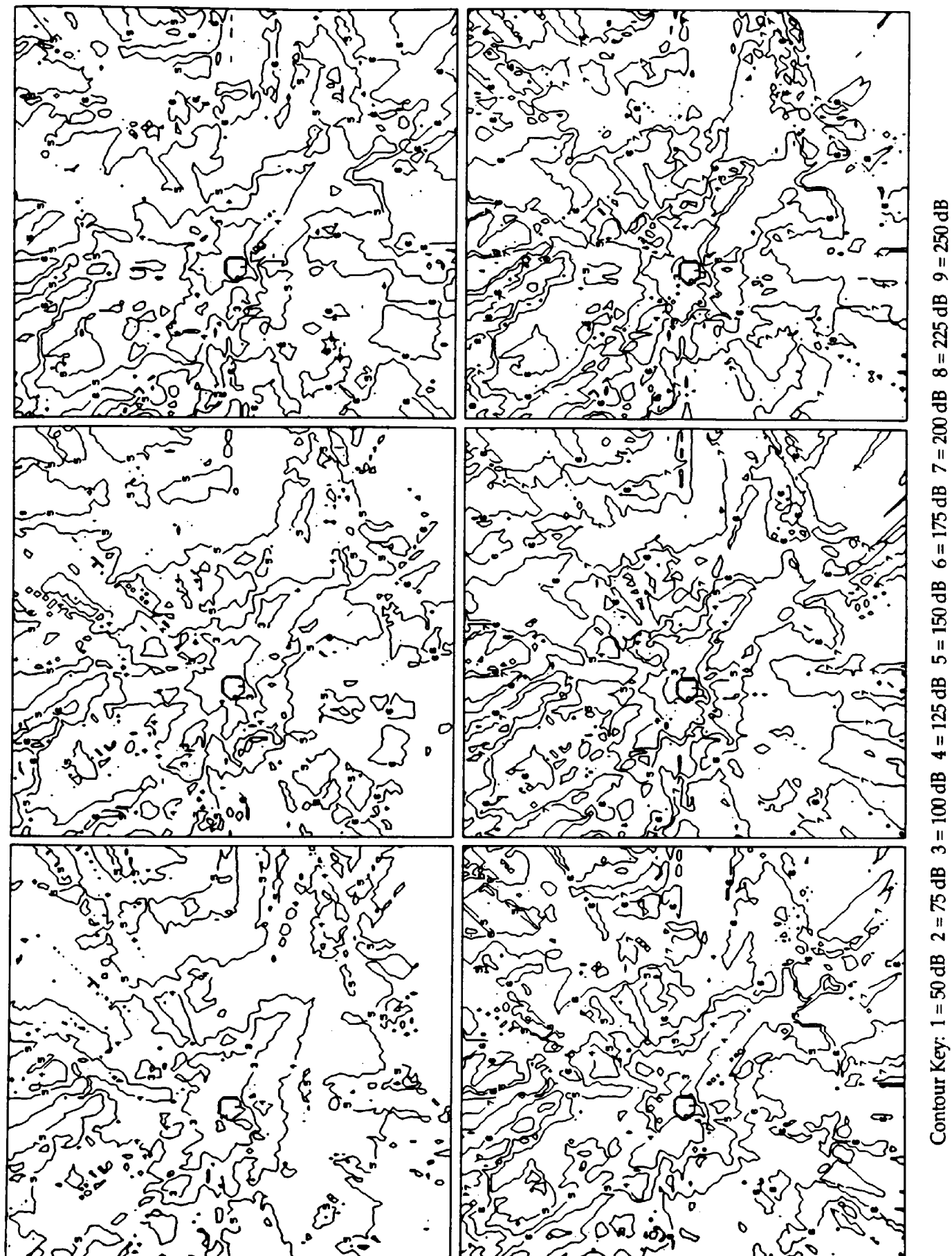


Figure B.10 - ST08 Field Losses (in dB) For a Base Station at (146,199)
Transmitting at 200, 400, 600, 900, 1400 and 1800 MHz.

Appendix C

Critical Surface-Specific Points

The results and maps presented here illustrate the output from procedures to identify critical points of a regular grid digital elevation model. Section 8.2 of Chapter Eight describes the fundamentals of each of these algorithms. Douglas (1986) gives a detailed description of some of the various algorithms for detecting critical points and information-rich linear features from regular grid DEMs. The primary advantage of selecting a subset of the regular grid is the extensive storage saving that may be attained when used as an irregular point DEM. The redundant data of the regular grid is identified and eliminated at a local level, such that the carefully selected points are finely adapted to the surface morphology.

Douglas Algorithm (1986)

Table C.1 represents the results of the Douglas algorithm applied to the 401×401 data sets ST06 and ST08. The table also includes the results for smoothed data sets (using a binomially weighted filter, as in Figure 8.3 of Chapter Eight). This filter removes some of the 'noise' within the data. Figures C.1 and C.2 illustrate the selected ridge and channel points (including saddles) for the unsmoothed and smoothed data set of ST08.

401×401 Data Set	Ridge Points		Channel Points		Saddle Points		Total Points	
ST06	5998	3.73%	6484	4.03%	26	0.016%	12508	7.78%
ST06 Smoothed	2814	1.75%	3130	1.95%	6	0.004%	5950	3.70%
ST08	10346	6.43%	11510	7.16%	91	0.057%	21947	13.65%
ST08 Smoothed	5638	3.51%	6587	4.10%	15	0.009%	12240	7.61%

Table C.1 - Number and Percentage of Selected Grid DEM Points for ST06 and ST08 (Douglas).

A triangulated irregular network (TIN) for the Douglas derived data set of ST08 (ie. row three in the above table) is shown in Figure C.7. The storage costs and errors associated with this TIN are presented along with Figure C.7.

Jenson Algorithm (1985)

The Jenson algorithm was applied to ST06 and ST08, together with the modified filter for 5×5 pixels. This prevents small elevation fluctuations in the terrain being identified as ridges or channels. The algorithm is illustrated in Figure 8.1 of Chapter Eight. The results for both of these filters applied to the original data sets and binomially smoothed data sets are presented

in Table C.2. Figures C.3 and C.4 illustrate the selected points for the 3x3 and 5x5 filters of the original data set for ST08. A comparison of these results (Table C.2) with those for Douglas' algorithm (Table C.1), together with the plots for the original algorithms and data sets, (Figures C.1 and C.3) would suggest that Douglas' algorithm tends to produce the most critical surface-specific points. However, the ridges and channels are not as continuous as for Jensen's algorithm, due to the latter's tendency to select a greater number of grid vertices.

401x401 Data Set	Ridge Points		Channel Points		Saddle Points		Total Points	
ST06 3x3	11671	7.26%	10752	6.69%	149	0.093%	22572	14.04%
ST06 Smoothed	7180	4.47%	6602	4.11%	38	0.024%	13820	8.59%
ST06 5x5	5646	3.51%	4920	3.06%	15	0.009%	10581	6.58%
ST06 Smoothed	5013	3.12%	4553	2.83%	11	0.007%	9557	5.94%
ST08 3x3	21136	13.14%	19289	12.00%	421	0.262%	40846	25.40%
ST08 Smoothed	14192	8.83%	13040	8.11%	112	0.070%	27344	17.01%
ST08 5x5	12435	7.73%	11195	6.96%	53	0.033%	23683	14.73%
ST08 Smoothed	11353	7.06%	10279	6.39%	54	0.033%	21686	13.49%

Table C.2 - Number and Percentage of Selected Grid DEM Points for ST06 and ST08 (Jenson).

For the purpose of selecting vertices for an irregular grid DEM, such as a TIN, both the Douglas and Jensen algorithms require further refinement. Additional procedures are needed to identify and generalise redundant points within the selected subset of grid vertices. However, the relative importance of points cannot be determined without an appraisal of their overall contribution to the final surface model within the DEM or TIN. As a result, generalisation algorithms tend to produce unconstrained DEMs which may produce large elevation errors, particularly in areas of steep slopes. This was illustrated in Figure 8.4 of Chapter Eight. The TIN in Figure C.7 includes some large errors, despite all of the selected vertices from Douglas' algorithm being incorporated. This proves the unsuitability of such algorithms for deriving grid points from a grid DEM for use in a TIN. However, the detected linear features could be used in conjunction with other point selection methods to form a constrained triangulation.

Makarovic Algorithm (1983) and Chen & Guevara Algorithm (1987)

The basic principle of both algorithms is the identification of critical points within the terrain, irrespective of the surface features that they represent. In general, both algorithms pass a spatial differential or 3x3 Laplacian operator over the data set to identify these critical points. However, whilst the calculation of these points may be quite simple, the user has no way of determining the optimum number of points for a DEM, such as a TIN. The user is also left with the problem of a DEM which is not constrained by maximum absolute error. Tables C.3 and C.4 show the number of selected points from data sets ST06 and ST08 respectively, for absolute

tolerances of 5 metres to 25 metres (the tolerance is calculated as the sum of the deviations from the centre cell of a 3x3 matrix to each of its eight grid neighbours). Figures C.5 and C.6 illustrate all the critical points within ST06 and ST08 that cannot be predicted to within ± 5 and ± 10 metres respectively, using this selection criterion. However, to construct an irregular point DEM or TIN, these critical points need to be supplemented with vertices of very low frequency or tolerances. To illustrate this, consider Figure C.6. It can be seen that the selected high frequency points of the Taff valley in the region of 240,110 to 170,170 are mainly made up of ridge points on either side of the valley. A corresponding TIN of these vertices will result in triangle edges crossing this valley, creating large elevation errors.

Tol. (m)	Points	%	Tol. (m)	Points	%
25	268	0.17	14	1682	1.05
24	303	0.19	13	2004	1.25
23	355	0.22	12	2433	1.51
22	411	0.26	11	2940	1.83
21	491	0.31	10	3576	2.22
20	587	0.37	9	4404	2.74
19	704	0.44	8	5525	3.44
18	827	0.51	7	6995	4.35
17	978	0.61	6	8943	5.56
16	1154	0.72	5	11614	7.22
15	1385	0.86			

Table C.3 - Selected Points from 3x3 Laplacian Filter of ST06.

Tol. (m)	Points	%	Tol. (m)	Points	%
25	1181	0.73	14	7032	4.37
24	1371	0.85	13	8403	5.23
23	1598	0.99	12	10124	6.30
22	1880	1.17	11	12185	7.58
21	2204	1.37	10	14617	9.09
20	2556	1.59	9	17717	11.02
19	3011	1.87	8	21460	13.35
18	3557	2.21	7	26280	16.34
17	4231	2.63	6	32156	20.00
16	4981	3.10	5	39779	24.74
15	5905	3.67			

Table C.4 - Selected Points from 3x3 Laplacian Filter of ST08.

An example of the output from an algorithm which detects high and low frequency data derived from passing a 3x3 Laplacian filter through the data set is illustrated in Figure C.8, whilst its corresponding TIN is shown in Figure C.9. The errors associated with this DEM are less than those for that of the Douglas TIN, but may still be considered intolerable. As such, TINs of critical terrain points or features are neither error-constrained or storage-efficient.

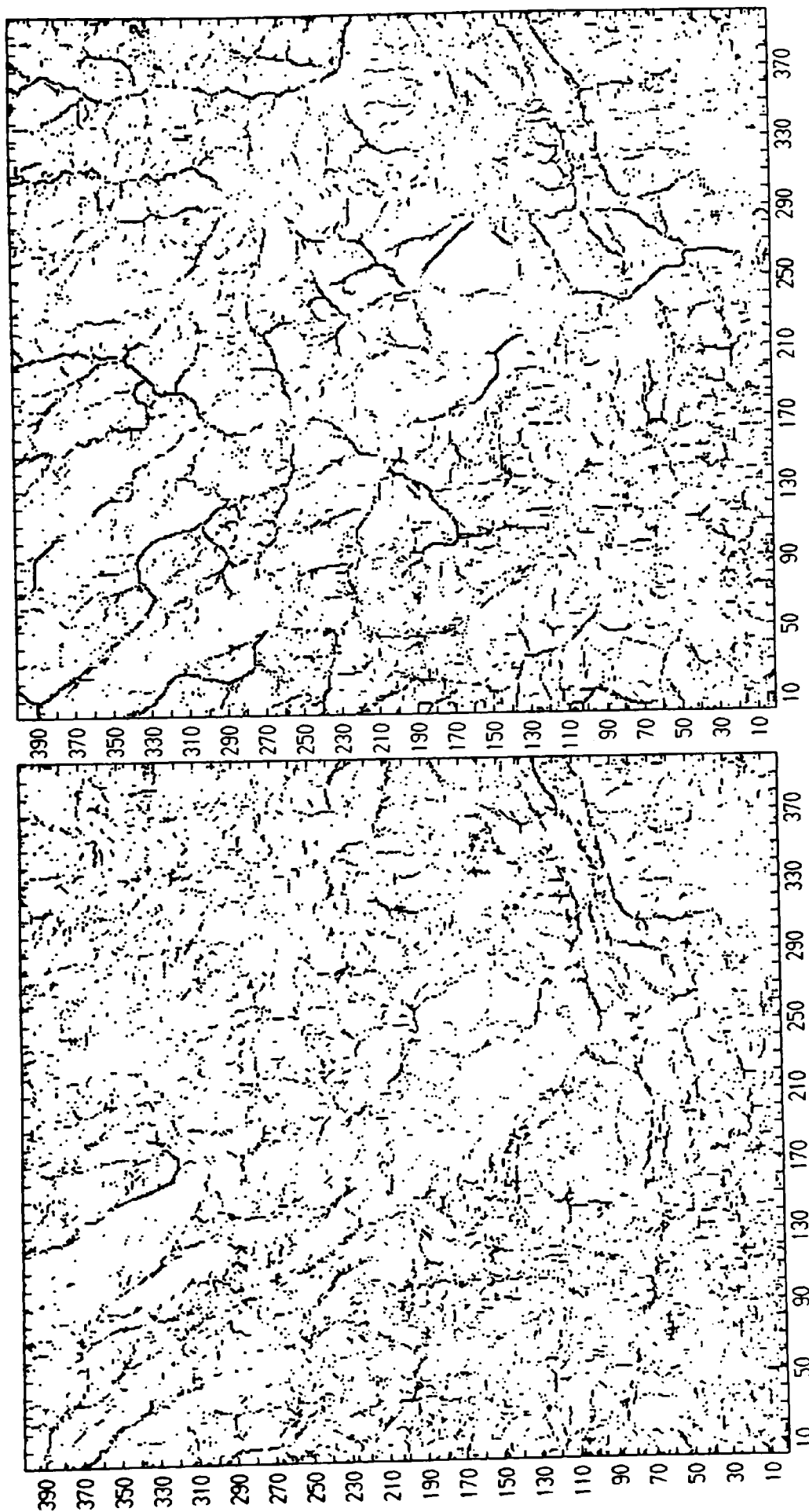


Figure C.1 - Douglas Algorithm Applied to Data Set ST08
 (a) Ridge Points (b) Channel Points

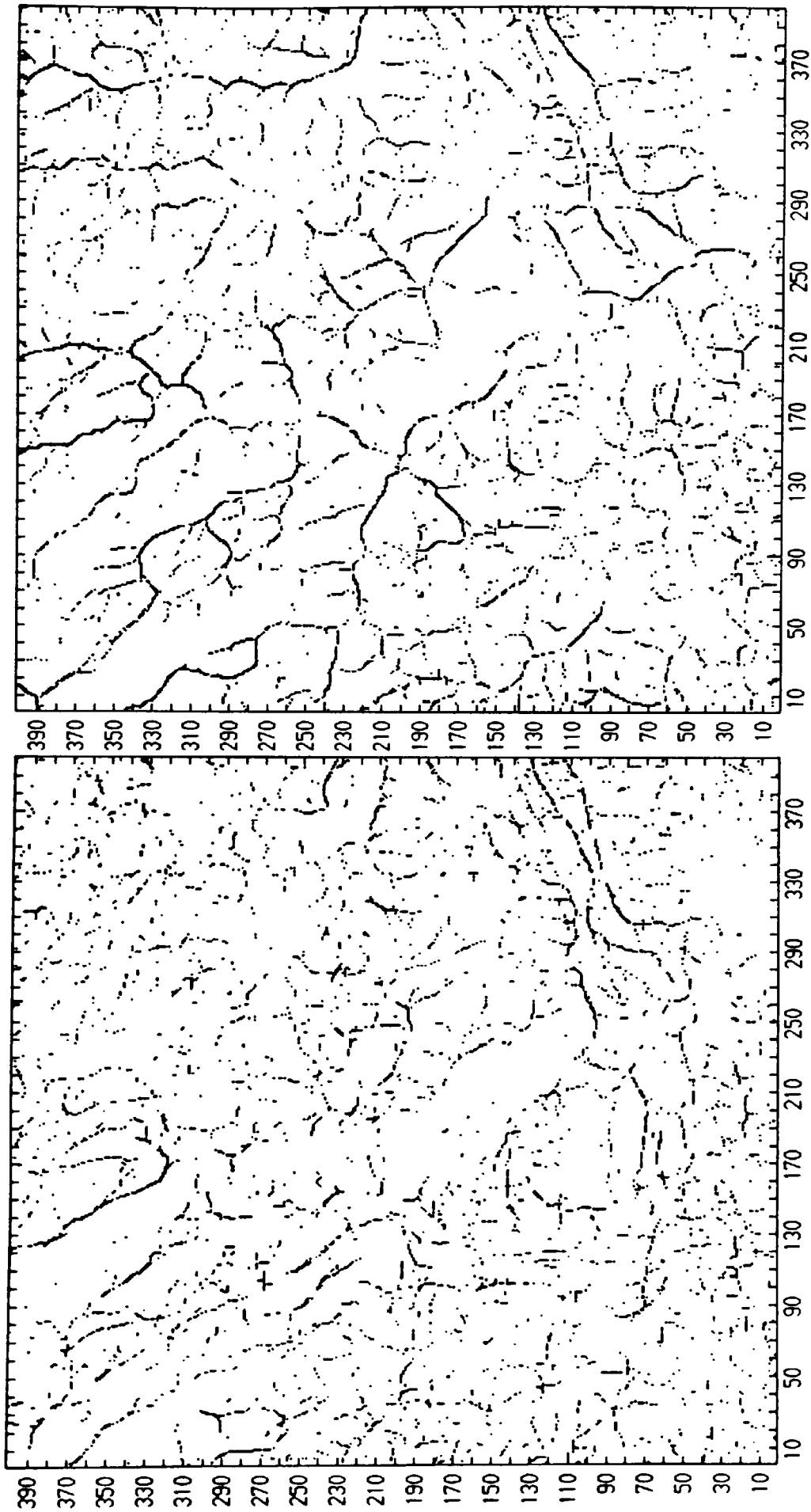


Figure C.2 - Douglas Algorithm Applied to Smoothed Data Set ST08
 (a) Ridge Points (b) Channel Points

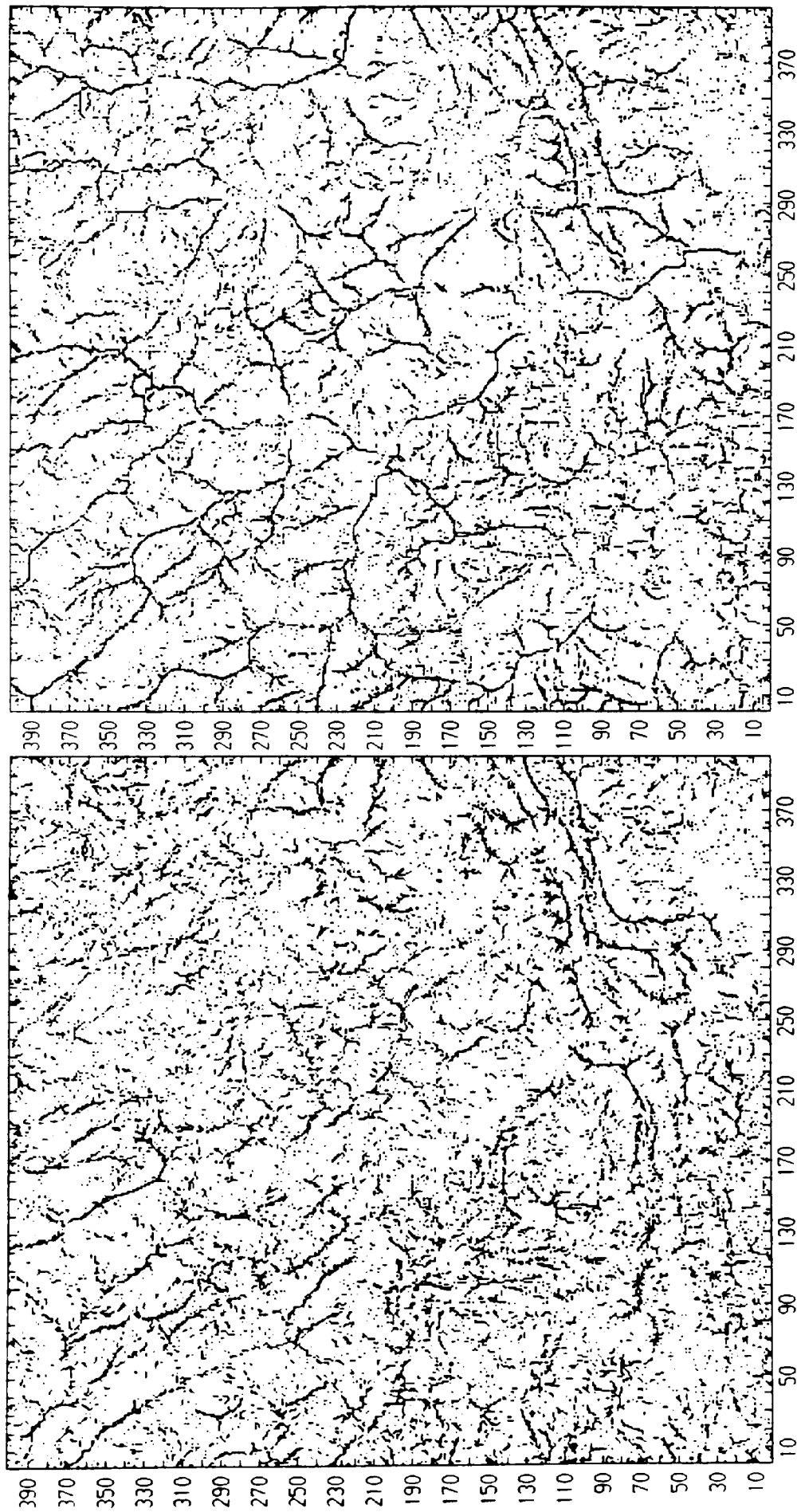


Figure C.3 - 3x3 Jensen Algorithm Applied to Data Set ST08
 (a) Ridge Points (b) Channel Points

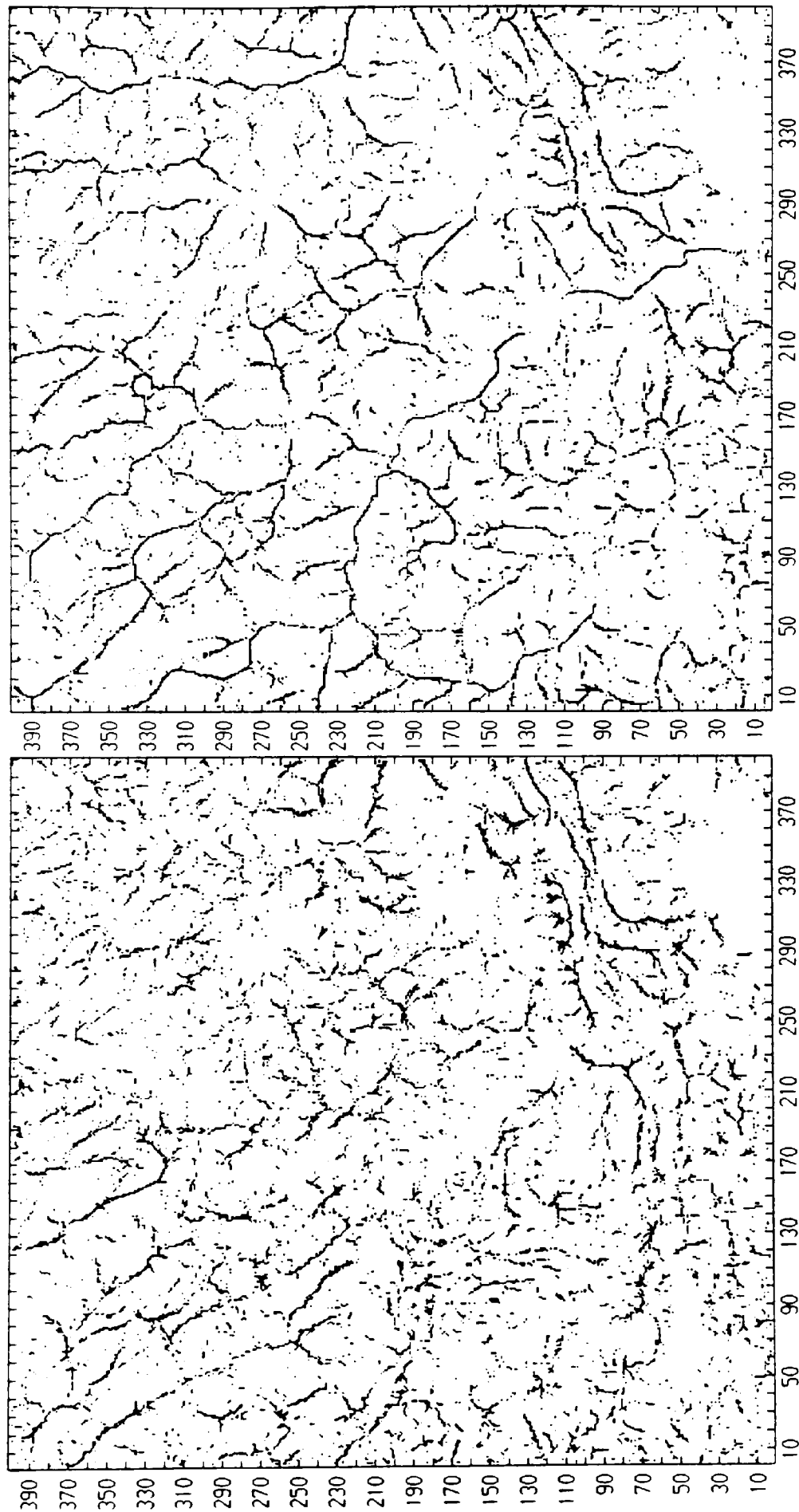


Figure C.4 - 5x5 Jensen Algorithm Applied to Data Set ST08
 (a) Ridge Points (b) Channel Points

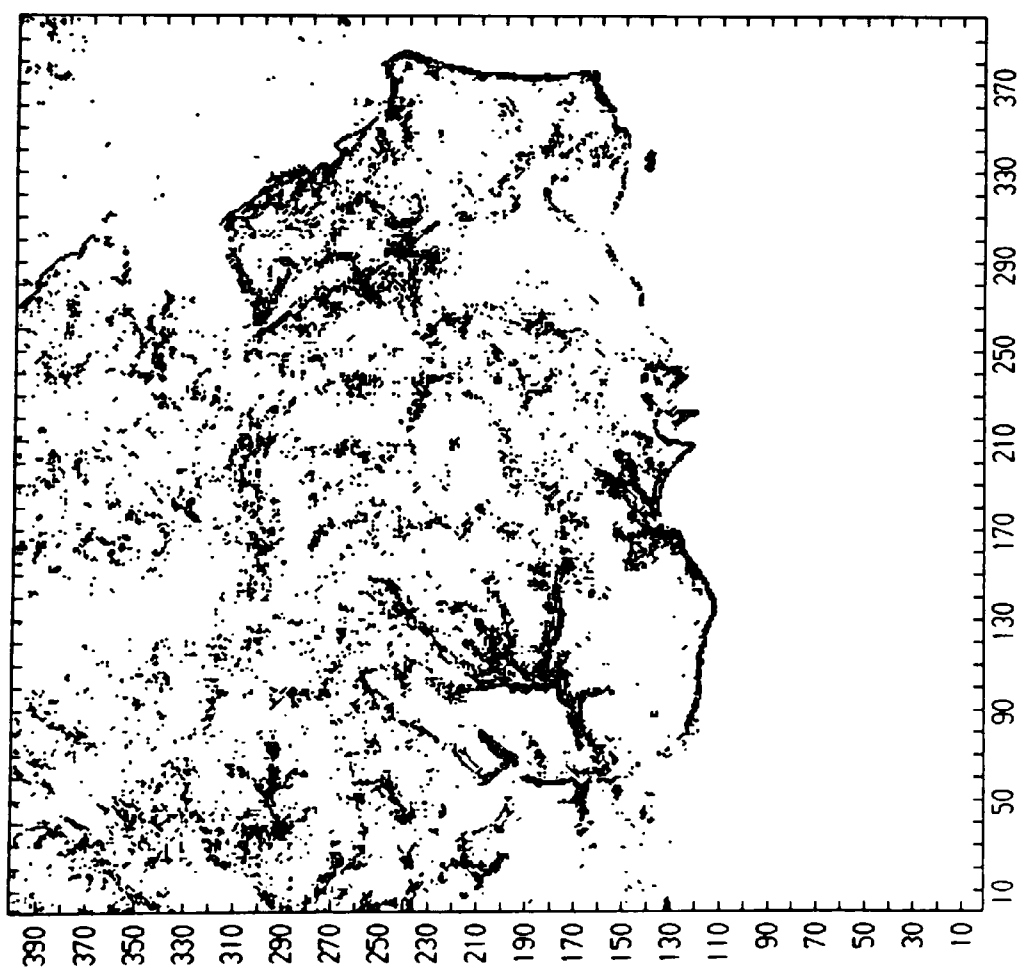


Figure C.5 - ST06 Critical High Significant Points within Laplacian Tolerance of 5 metres

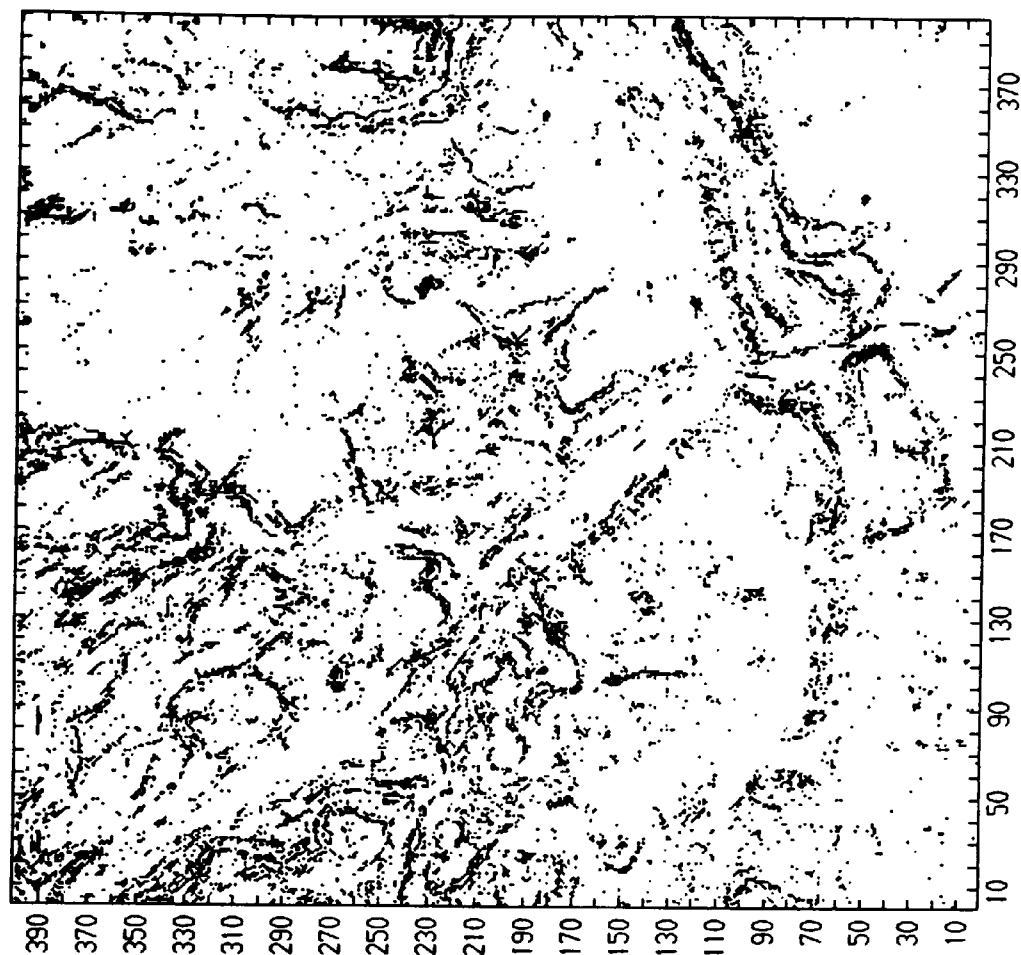
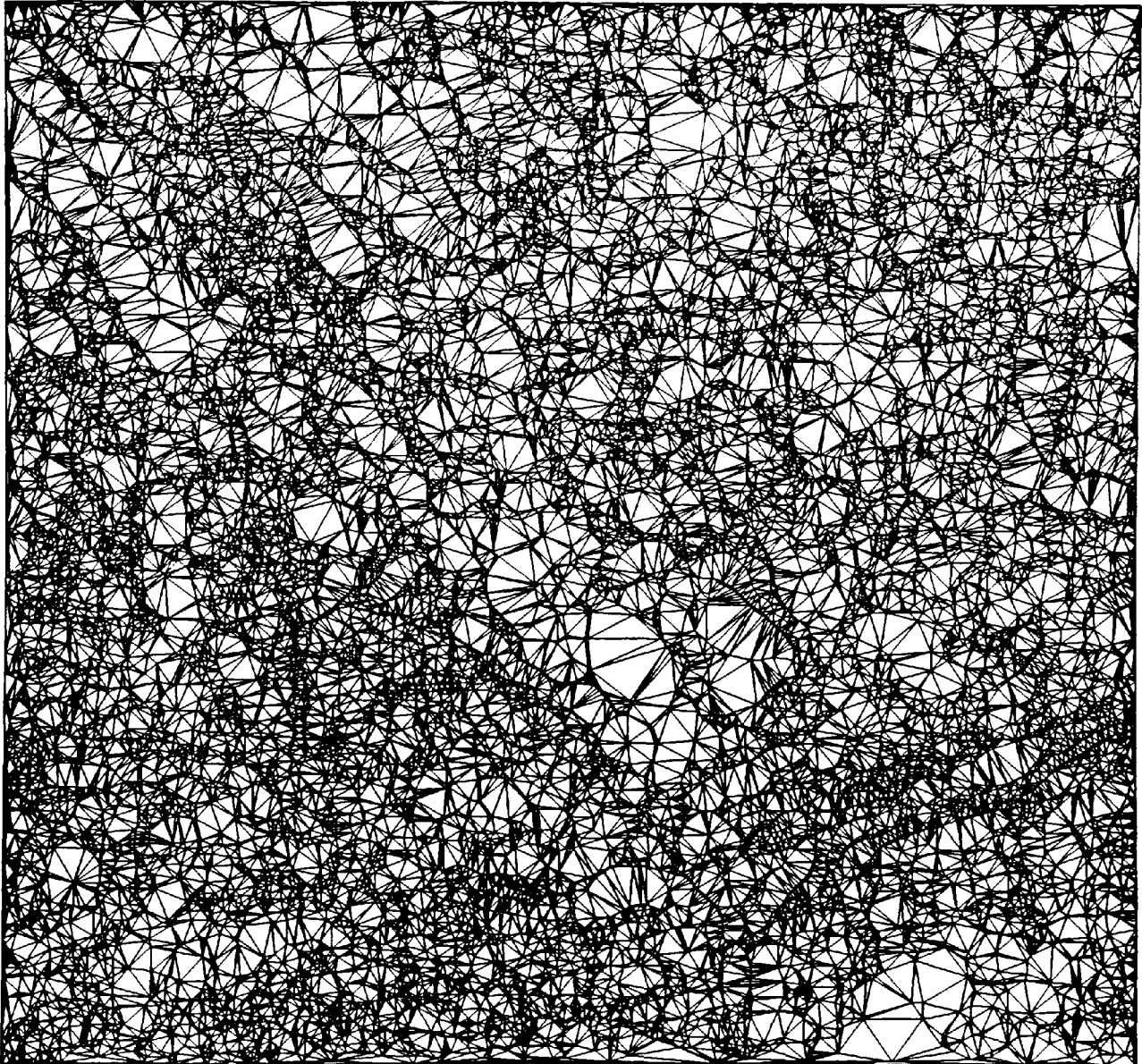
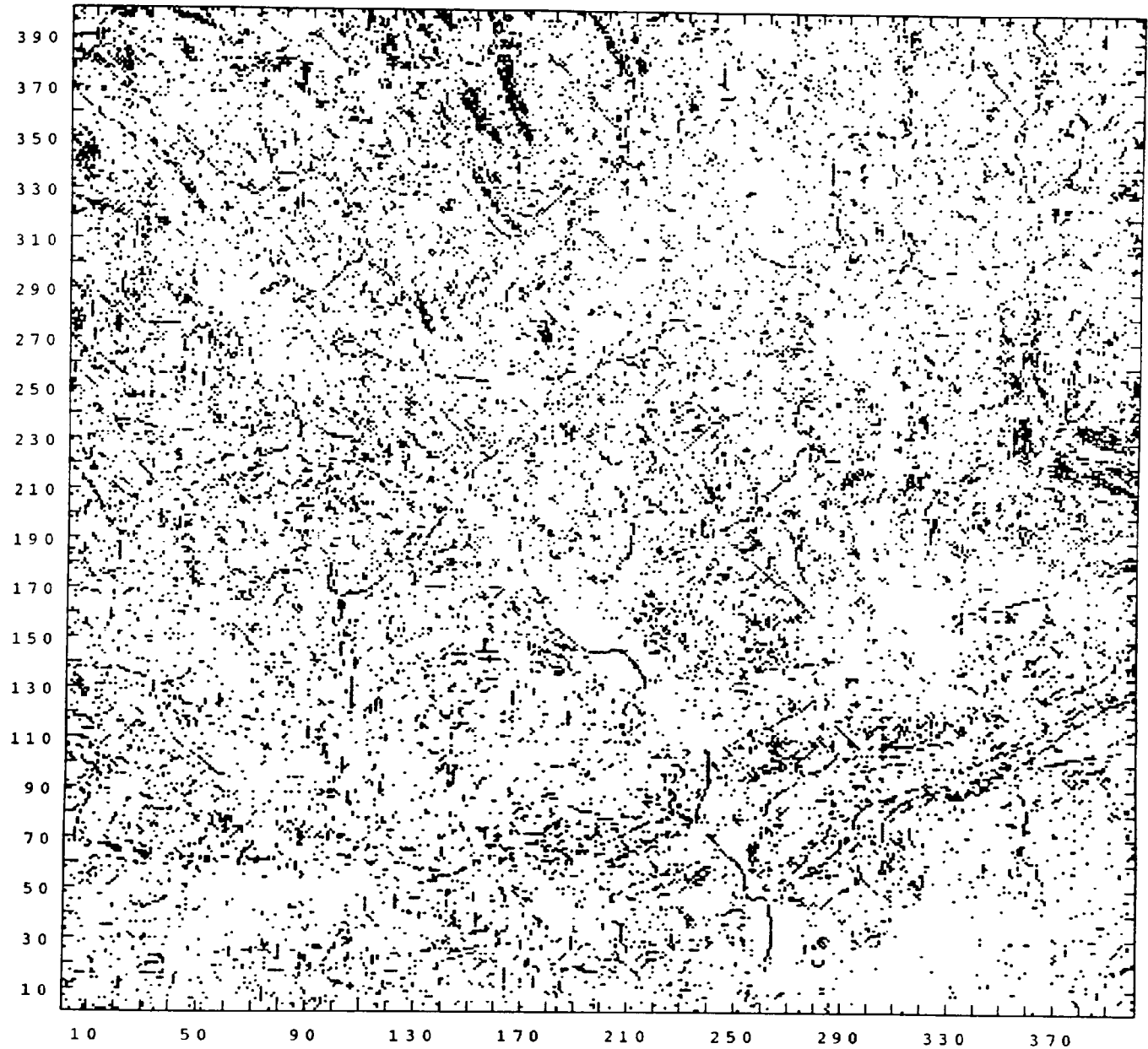


Figure C.6 - ST08 Critical High Significant Points within Laplacian Tolerance of 10 metres



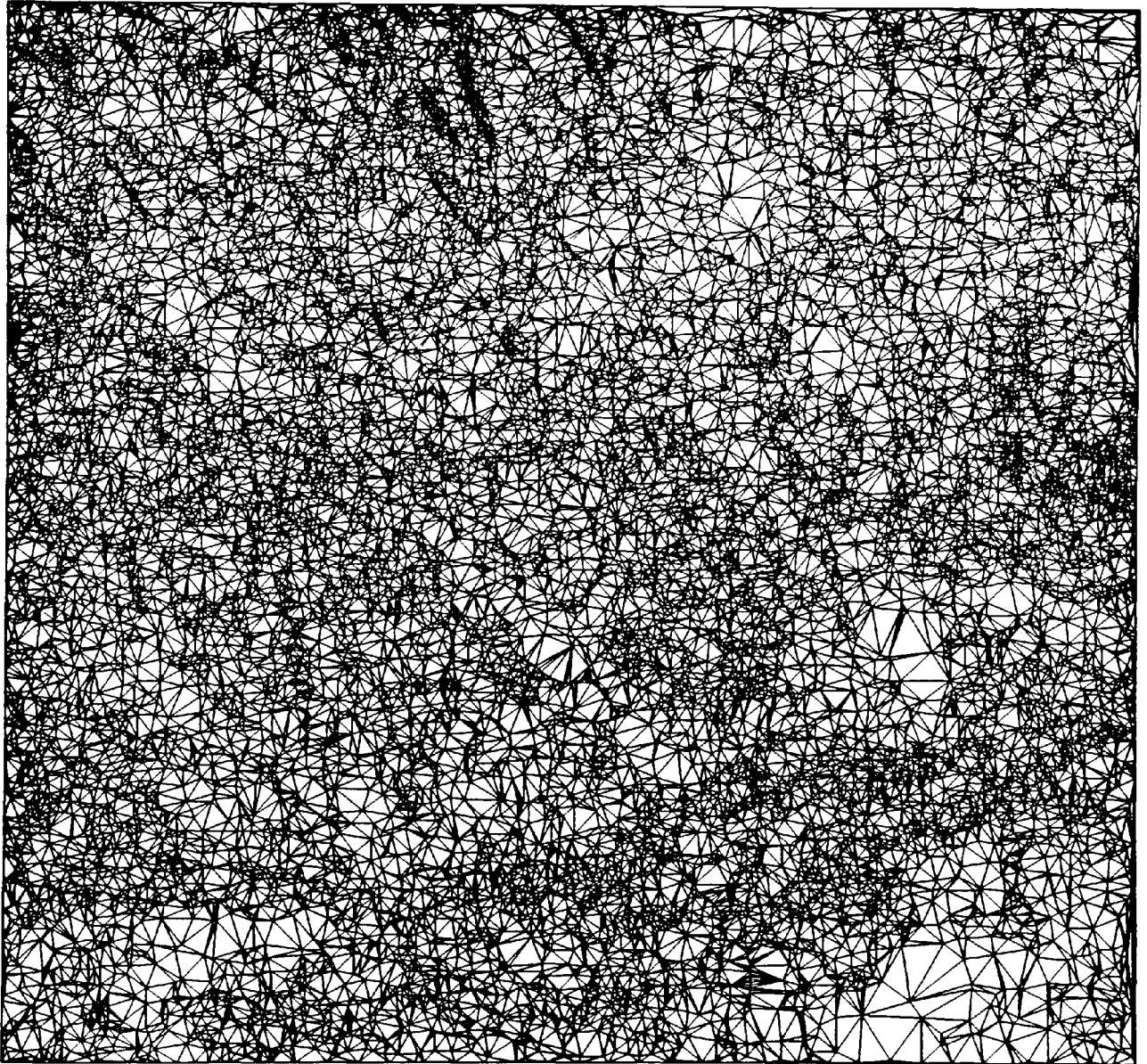
Ridge Points	= 10,346	Triangles	= 44,303
Channel Points	= 11,510	Edges	= 133,322
Saddle Points	= 91	% Storage	= 138.53%
Boundary Points	= 413	Absolute Average Error	= 3.542 m
Total Points	= 22,359	Standard Deviation	= 7.449 m
% of Grid Points	= 13.91%	Root Mean Square Error	= 6.659 m

Figure C.7 - TIN of Douglas' Selected Points for ST08 (as in Figures C.1a & C.1b).



High Frequency = 10,098; Low Frequency = 9,880; Total = 19,978 (12.42%)

Figure C.8 - Selected Very Important Points (VIPs) for ST08.



High Frequency	= 10,098	Triangles	= 40,263
Low Frequency	= 9,880	Edges	= 121,098
Boundary Points	= 309	% Storage	= 125.77%
Total Points	= 20,287	Absolute Average Error	= 2.925 m
% of Grid Points	= 12.62%	Standard Deviation	= 5.411 m
		Root Mean Square Error	= 4.706 m

Figure C.9 - TIN of VIP Selected Points for ST08 (as in Figure C.8).

Appendix D

Huffman-Encoded Compression of O.S. Data Sets

The algorithm for compressing the regular grid data sets of ST06/08 (in Ch. Five) was applied to a further 41 O.S. data sets covering South Wales and Southern England. A statistical analysis of these data sets and the storage savings obtained are presented in Table D.1.

Data Set	Elevation Range	No.of Hgts.	Mean	Stan. Dev.	Code Range	No.of Codes	Av.Code Length	% Storage Saving
ST06	0 - 135	136	31.32	34.39	-40 - +41	52	1.968	87.70
ST26	0 - 108	109	1.90	7.95	-59 - +51	62	1.145	92.84
ST46	0 - 241	242	52.99	52.76	-74 - +71	84	2.282	85.74
ST66	0 - 238	239	96.34	52.91	-33 - +26	49	2.746	82.84
ST86	24 - 211	188	84.94	34.16	-24 - +18	42	2.100	86.87
SU06	68 - 295	228	161.46	40.92	-35 - +28	51	2.259	85.88
SU26	86 - 297	212	160.87	35.54	-15 - +17	31	2.473	84.54
SU46	40 - 280	214	111.65	28.99	-14 - +15	29	2.297	85.65
SU66	30 - 169	140	62.63	21.28	-13 - +14	25	1.879	88.25
SU86	15 - 136	122	56.30	24.18	-20 - +19	33	1.828	88.58
TQ06	0 - 82	83	21.91	12.51	-20 - +19	40	1.550	90.32
TQ26	0 - 188	189	43.93	39.39	-11 - +13	25	1.765	88.97
TQ46	0 - 235	236	68.80	50.73	-58 - +57	80	2.177	86.39
ST08	19 - 470	452	173.17	88.43	-39 - +49	59	3.238	79.76
ST28	0 - 472	473	82.90	101.27	-39 - +63	66	2.859	82.13
ST48	0 - 308	309	51.11	69.68	-44 - +60	71	2.165	86.47
ST68	0 - 250	251	74.10	54.07	-27 - +27	47	2.296	85.65
ST88	51 - 244	194	118.39	35.68	-20 - +25	43	2.014	87.41
SU08	60 - 180	121	96.16	17.02	-20 - +13	26	1.629	89.82
SU28	60 - 270	211	114.32	47.35	-17 - +21	35	1.909	88.07
SU48	40 - 236	197	88.99	44.39	-23 - +17	31	1.838	88.51
SU68	28 - 257	230	121.01	52.54	-15 - +16	32	2.469	84.57
SU88	19 - 216	198	97.90	46.19	-19 - +16	36	2.354	85.29
TQ08	13 - 155	143	64.16	27.75	-12 - +17	26	1.890	88.19
TQ28	0 - 144	145	44.76	32.78	-13 - +12	26	1.798	88.76
TQ48	0 - 113	114	35.93	29.12	-10 - +8	19	1.695	89.41
SO20	12 - 591	572	198.25	143.32	-35 - +32	63	3.189	80.07
SO40	0 - 302	303	116.43	65.37	-31 - +32	60	3.241	79.74
SO60	0 - 289	289	61.88	67.98	-42 - +52	55	2.512	84.30
SO80	0 - 297	298	148.27	79.52	-33 - +27	56	2.877	82.02
SP00	74 - 289	216	155.59	41.22	-21 - +17	37	2.303	85.61
SP20	63 - 242	180	109.78	35.67	-17 - +19	34	1.850	88.44
SP40	51 - 172	122	78.22	20.49	-15 - +17	30	1.739	89.13
SP60	51 - 251	201	82.80	21.06	-14 - +12	26	1.761	88.99
SP80	71 - 263	193	138.47	45.08	-71 - +71	82	2.164	86.47
TL00	57 - 243	187	121.72	31.17	-13 - +11	23	2.074	87.04
TL20	18 - 132	115	75.39	26.47	-14 - +15	29	2.060	87.13
TL40	26 - 117	92	72.12	15.04	-11 - +10	22	1.797	88.77
SO22	64 - 810	747	306.81	169.91	-35 - +43	67	3.443	78.48
SO42	17 - 422	401	101.53	49.35	-29 - +24	49	2.698	83.14
SO62	3 - 326	318	75.27	46.63	-74 - +78	52	2.752	82.80
SO82	0 - 330	331	47.82	56.68	-19 - +16	34	1.980	87.62
SP02	24 - 211	188	84.94	34.16	-24 - +18	42	2.100	86.87

Table D.1 - Surface Statistics and Huffman-Encoded Storage Savings
for 43 O.S. 401x401 Regular Grid Data Sets.

To compact a DEM into a storage-efficient format, it is beneficial to know in advance the level of storage saving that can be attained for any given data set. This could be achieved by 'characterising' terrain, but no single surface statistic exists which can fulfil this requirement. The general approach is to consider a number of statistics, such as relief, slope, aspect, standard deviation, convexity, etc. The relationship between the standard deviation of elevations for each data set, with respect to storage saving is illustrated below in Figure D.1.

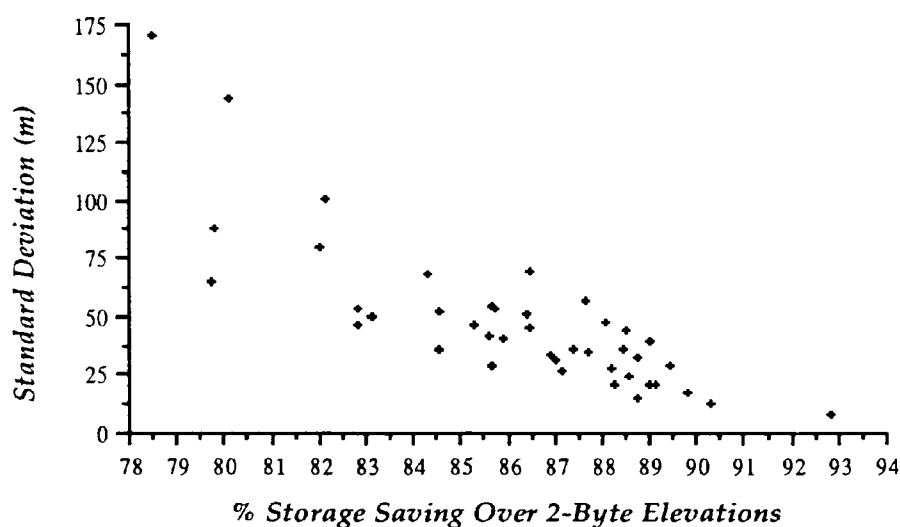


Figure D.1 - Relationship between DEM Standard Deviation and Storage Saving Possible using a Linear Prediction Huffman-Encoding Algorithm.

As expected, there is no distinct relationship between the standard deviation and the degree of data compression possible. A better approach for characterising terrain (in order to determine the level of storage saving possible), is to relate the surface statistic to the method of data compression. For the algorithm used in this study, deviations from a linear predicted trend are Huffman-encoded. Hence, a useful surface-statistic would be the average absolute deviation from this trend (measured in metres). The relationship between this statistic and storage saving for the 43 O.S. data sets is illustrated below in Figure D.2.

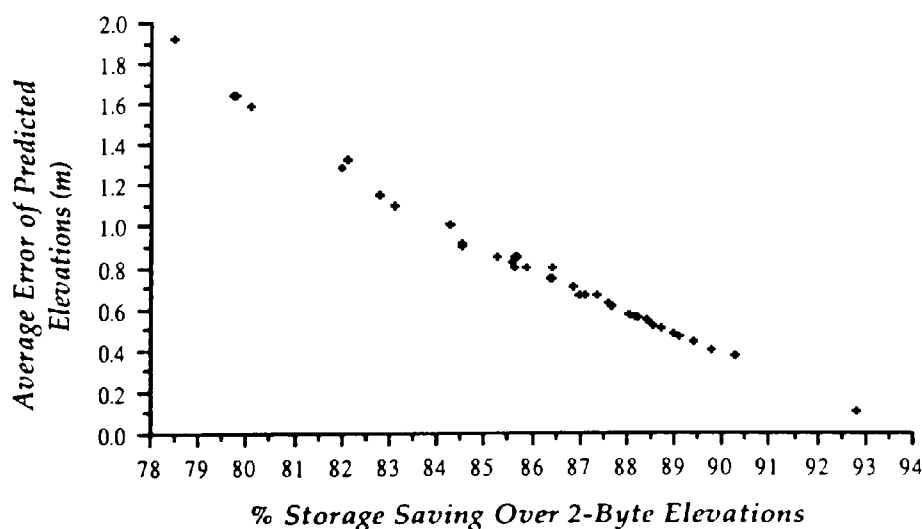


Figure D.2 - Relationship between Average Deviation and Storage Saving.

Figure D.2 illustrates the strong correlation between average deviation and storage saving. This relationship only holds for data derived from regular grid data sets measured to a vertical resolution of one metre, such as for O.S. and U.S.G.S. data sets. However, in order to determine the possible storage saving of a regular grid DEM, it is important that this surface statistic (average deviation) can be determined from a random sample of elevations. The stability of this statistic was examined by randomly sampling 1% of grid points (1600) for each of the 43 data sets and calculating the average deviation. The graph of these results is illustrated below in Figure D.3.

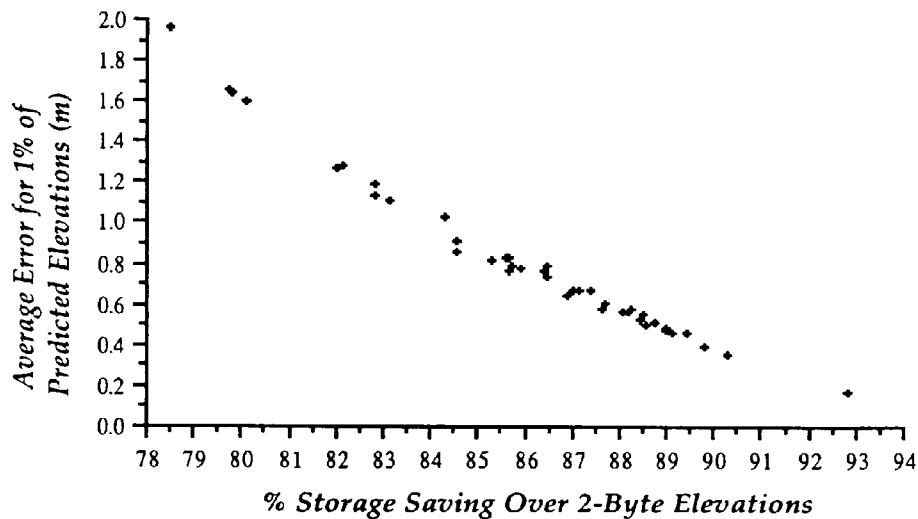


Figure D.3 - Relationship between Ave. Deviation of a 1% Random Sample and Storage Saving.

Figure D.3 is almost identical to that of of Figure D.2, even though it has been derived using only 1% of the original data. As a result, the storage savings of all the data sets can be accurately predicted to within $\pm 1\%$ by fitting a simple least squares line through the original data (Figure D.2). The benefits of such an approach for characterising terrain will be of benefit for users wishing to optimise disk storage for DEMs. As such, the degree of data compression can be determined for any data set within a few milliseconds.



**The Polytechnic of Wales
Politechnig Cymru**

**DIGITAL TERRAIN MODELS
FOR
RADIO PATH LOSS CALCULATIONS**

(Supplement - Papers)

David B. Kidner

Department of Mathematics and Computing

**A thesis submitted in partial fulfilment of the requirements of the
Council for National Academic Awards (CNAA)
for the degree of Doctor of Philosophy.**

November 1991

DIGITAL TERRAIN MODELS FOR RADIO PATH PROFILES *

David B. Kidner, Christopher B. Jones, David G. Knight, Derek H. Smith
Department of Mathematics and Computing
The Polytechnic of Wales
Pontypridd, Mid Glamorgan
Wales, CF37 1DL, U.K.
e-mail: DBKidner @ UK.AC.POW.GENVAX

Abstract

Several methods of representing digital terrain models have been analysed in terms of storage space, errors and speed of access, for the purpose of retrieving profiles for radio path loss calculations. When compared to a regular grid, space reductions of at least 70% can usually be achieved with constrained triangulated networks and surface patch quadrees, with maximum errors within 2% of elevation range. Unconstrained methods of polynomial grids, sub-sampled grids and grid cell approximations give similar space savings with average error under 1%. Error-free data compression techniques can then be applied to representations such as sub-sampled grids to result in a 90% space saving. For fast access, regular grids and polynomial grids are significantly superior.

Introduction

Optimal positioning of radio transmitters and receivers requires the generation of numerous terrain profiles for use in path loss prediction algorithms. Rapid retrieval of profiles from very extensive digital terrain models (DTMs) raises the question of determining the most efficient means of storing terrain data. There are many techniques available for representing terrain models. Consequently, there is a requirement for comparative studies of efficiency which can assist in discriminating between modelling techniques on the basis of application-specific criteria. For the purpose of radio path profile extraction, the major issues include minimising storage space, controlling errors in elevation, and speed of profile generation. The first two of these issues are clearly of relevance to many other applications of DTMs, such as, for example, intervisibility studies, visual simulation of landscape and prediction of flooding due to sea level changes.

In many terrain modelling applications there is a choice between selecting a model and then testing its performance for the specific application, or alternatively adopting a new structure specifically aimed at the requirements of the application. These conflicting approaches have raised questions regarding a model's flexibility and efficiency. This dilemma of adopting or designing the DTM has arisen within this programme of work. It was decided to investigate a number of existing methods for representing terrain including the regular grid, triangulated irregular network (TIN), surface patch quadtree, polynomial surfaces, and alternative grid methods. This was to be accomplished with regard to their storage efficiency and error performance, together with an analysis of

*This work has been carried out with the support of Procurement Executive, Ministry of Defence

their suitability for profile extraction (error and time efficiency) and in particular for path loss estimation. From these results it was hoped that the requirements of an efficient, application-specific DTM could be identified and hence implemented. This paper covers the first part of this work, ie. an investigation of existing, popular DTMs.

Choice of DTM

DTMs are used for representing terrain in the form of surfaces that can be mathematically or numerically defined. They are broadly classified into models which are based on structuring the points into some specific order, taking into account their spatial relationships, and models which are based on fitting mathematical functions to the elevation data. To estimate the radio path loss between possible transmitter and receiver locations, the profile needs to be retrieved from the DTM, and a path loss algorithm applied.

At present, the most common approach to storing terrain data is in the form of a regular grid of elevations or digital elevation model (DEM). This is stored as a matrix, such that the position of points in the structure is implicit. Thus, most applications are very time-efficient, since search time is minimal. However, since the terrain is modelled at a relatively fine resolution throughout (so that all significant features are incorporated), there are high storage costs and excessive data redundancy in uniform regions of terrain. Several models were considered as alternative, storage-efficient methods for overcoming these problems.

The TIN (Peucker et al, 1978) is one of the most popular alternatives to the regular grid, since all data points are honoured directly to form the vertices of the triangles which are used to approximate the terrain (Petrie, 1987). The use of triangles offers a relatively simple way of incorporating significant terrain features, such as peaks, pits, ridges, channels and breaklines. This might also be considered important for the current application of radio path-loss, since locating the obstructions that will interfere with the transmission will be critical (Edwards and Durkin, 1969).

The surface patch quadtree (Chen & Tobler, 1986; Leifer & Mark, 1987) recursively divides a grid DEM into quadrants or sub-quadrants, which are approximated by mathematical functions. At any level of representation, the patch (sub-quadrant) will approximate the terrain to within a user-specified error-tolerance. The variable size of the square patches ($2^n \times 2^n$ pixels) allows the method to adapt to the terrain.

Mathematical representations of terrain have received widespread attention, particularly in the Earth sciences (Davis, 1986; Krumbein & Graybill, 1965). Polynomials (and power series) are the most consistent, since any continuous surface can be approximated satisfactorily by a polynomial of sufficiently high degree. In general, the trend of the terrain can be easily identified by a low-order polynomial. However, to ensure consistency and adaptability, polynomials are best applied locally.

Another method (or group of methods) examined consists of reducing the storage requirements of the original grid DEM. This was investigated in a

number of ways, including a reduction in the sampling interval, quantising the range of values as a step function (Boehm, 1967), together with some other data compression techniques, including Huffman coding.

This study has been based on the assumption that the source data for these prototype DTMs are initially in the form of a regular grid DEM of height values.

Prototype DTMs

TIN

A TIN is a terrain model that uses a surface of continuous, connected triangular facets, typically based on a Delaunay triangulation of irregularly spaced nodes or observation points (Burrough, 1986). Descriptions of algorithms to compute this are given for example by McCullagh & Ross, (1980) and by Sibson, (1978). The Delaunay triangulation is considered the best, since the triangles will approach equiangularity, and will be unique for an irregular set of points. Ideally, the links forming the network should represent critical terrain features, such as ridges, channels and breaklines.

There are a number of algorithms for extracting critical features from regular grid DEMs (Peucker & Douglas, 1975; Jenson, 1985; Douglas, 1986; and Chen & Guevara, 1987). These methods were each applied to grid DEMs, but all tended to produce superfluous data amongst the critical features (eg. ridges and channels). Thus post-processing of this data was needed to reduce the number of nodes to an 'acceptable proportion' of regular grid nodes. However, there was no guarantee that large errors would not occur after triangulation, between interpolated points and the original grid nodes. The problem is accentuated by the fact that what is regarded as an 'acceptable proportion' of grid nodes cannot be predetermined without detailed analysis of the terrain.

For our TIN, a sparse set of critical features, including peaks, pits, passes and other significant pixels or VIPs (Chen & Guevara, 1987) was selected. An initial triangulation was then constructed (in a similar manner to Fowler & Little, 1979; DeFloriani et al, 1983), which was refined locally by the insertion of new grid points until a user-specified error tolerance of the TIN was attained. It should be noted that there is a likelihood that points extracted from a grid will produce a degenerate triangulation. This occurs when more than three cells of the Thiessen or Voronoi diagram (dual of the Delaunay triangulation) meet. Thus degenerate cases had to be identified and avoided.

The data structure of the TIN is based upon Peucker et al, (1978), whereby the nodes are considered as the primary entities of the database (as opposed to the triangles themselves). The coordinates of each node are stored, together with a pointer to a list of neighbouring nodes, the average number of which will be six. A list of triangles associated with each directed edge could also be stored, but the node and pointer list will be sufficient for most applications.

Profiling is achieved by linearly interpolating points at the intersection of the profile with the triangle edges (ie. at the change in slope). Thus the algorithm traces through the network of edges (found from the neighbour list of each node), interpolating the profile points, from triangle to triangle.

Surface Patch Quadtree

The surface patch quadtree overcomes the grid DEM data redundancy problem by adapting to the variability in the terrain. This is accomplished by recursively sub-dividing the original DEM into quadrants and sub-quadrants, which are approximated by mathematical surface functions. If any of the original grid heights cannot be re-interpolated to within a specified error tolerance, the particular quadrant will be sub-divided further. Alternatively, the user could opt for a more general sub-division criterion, such as average error or RMSD. Uniform topography can be represented by a small number of large patches, whilst variable terrain can be approximated by a greater number of small patches.

A storage-efficient method of representing the quadtree can be achieved in the form of a linear quadtree (Gargantini, 1982), thus avoiding the need to store pointers. In the surface patch quadtree, every node that is stored will be 'black', (as opposed to black or white in conventional quadtrees), since every node is a patch that approximates the surface within that square region.

The surface patch quadtree implemented in this study is based upon the linear quadtree of Gargantini, but applied in a manner similar to that of Chen and Tobler. The sub-division of the quadtree is based upon maximum absolute deviation. A number of mathematical surface functions were tested, but the ruled surface (hyperbolic paraboloid) produced the best overall results, as in Chen and Tobler's survey. The form of this equation is $f(x,y) = ax+bx^2y+cy+e$, which is fitted to the four corner elevations of the normalised patch, such that

$$Z_{ij} = a*(i-i_1)+b*(i-i_1)(j-j_1)+c*(j-j_1)+e, \quad \text{where } x = i-i_1 \text{ and } y = j-j_1$$

$$\begin{array}{lll} Z_1 = f(0,0) & = & e \\ Z_2 = f(1,0) & = & a+e \\ Z_3 = f(0,1) & = & c+e \\ Z_4 = f(1,1) & = & a+b+c+e \end{array} \quad \begin{array}{ll} a = Z_2-Z_1 \\ b = Z_4+Z_1-Z_2-Z_3 \\ c = Z_3-Z_1 \\ e = Z_1 \end{array}$$

The main advantage of this and similar functions, is that the surface can be represented by five 2-byte integers, (the address of the quadtree node and the four elevations of the patch corners). This is based on the assumption that the address or key can be compressed into a two-byte integer that will denote the size and position of the surface patch. In this study, grid DEMs of 256x256 pixels were used, so the key represents patches of between 2x2 and 128x128 pixels, ie. seven levels of representation.

Profiles are retrieved from the quadtree by creating a list of intersected nodes (patches), starting from the root node. If a patch contains a section of the profile, a search is made of the quadtree for that node, which if found can be used to interpolate the profile points. Otherwise, this current intersected node is sub-divided and the process recursively repeated. At each stage, the nodes that are not intersected can be omitted from the indexed search.

Polynomial Surface Model

Polynomials offer an effective, yet simple way of identifying the general characteristics of a surface. They are best applied locally, by a least squares fit of the data. Usually this data will be in the form of critical points of the terrain, but anomalies may occur if the data are too sparse in certain regions. Thus, a good

distribution of points, such as a regular grid, is essential. Polynomials can then be applied to equal-sized patches of this grid. Problems will occur at the boundaries of neighbouring patches, since surfaces are independent of each other, giving rise to discontinuities. This can be overcome by blending the patches together (Junkins et al, 1973). This will increase the storage requirements of the model, but is essential for applications such as contouring. For this study however, a continuous surface is not a prerequisite, so the patches were not blended. Since the polynomials are fitted to regular grids, the method is constrained to give the best possible fit throughout the whole patch, so the discontinuities are generally within an 'acceptable level' (± 10 m) for profiling.

Polynomials were tested for degrees one to ten, for patches up to 25x25 grid cells. In general, it was found that in terms of storage and error there is no distinguishable difference between polynomials of different orders, so low-order polynomials are usually recommended. Higher degree polynomials can occasionally be inconsistent, especially if the data are irregularly sampled. The coefficients for each surface can then be stored in a grid. Profiling is accomplished by calculating which patches are intersected by the profile and then simply substituting the profile coordinates into the relevant polynomials.

Other Grid Techniques

The main objective of this research is to produce a storage-efficient DTM, which can be used for calculating path losses. This can be achieved either by implementing a DTM, such as one of the above, or reducing the storage requirements of the original grid DEM. The latter can be achieved in a number of different ways.

One approach is based on increasing the grid sampling interval between points of the DEM. Since we are working from a 50 metre grid, this can be accomplished by deleting points from the grid to make it a 100, 150 or 200 metre grid, etc. Alternatively, the points comprising the new grid could be taken as a weighted average of the deleted points. This will obviously result in a loss of terrain features, the significance of which could not be estimated until the new grid was used to calculate the radio path losses for the test profiles. This method would have the added benefits of ensuring a faster profile extraction time than for the 50m grid, together with a storage saving of at least 75%.

A second approach is to store the grid as a number of microgrids or in a differential altitude representation (Boehm, 1967). This is accomplished by storing a sub-grid (say 5x5 or 9x9 pixels) in a compressed form which relates the difference between neighbouring heights as a two-bit flag representing either an increase, decrease or no change in elevation. This difference is stored as a constant for each microgrid, together with the lower left, reference elevation. Alternative representations could include one-bit flags for \pm constant, or a three-bit flag for representing $\pm 3 \times$ constant, inclusive. Thus a 5x5 grid can be represented in 8 bytes for a two-bit difference or 12 bytes for a three-bit difference.

A third approach is based upon compacting the data, but ensuring the original elevations are not changed. A typical grid of elevations may be stored as an array of two-byte integers, though this gives an excessive range (0-65,535). However, one byte (range 0-255) is usually insufficient. One alternative is to store an elevation in nine bits (range 0-511) and the offset of its neighbouring

elevation in the remaining seven bits (range -64 to +63). There are many other ways available for data compaction, such as Huffman coding (Williams, 1986).

Radio Path Loss Algorithm

The basic principle behind radio path loss algorithms is to estimate the shape of the profile between the transmitter and receiver, with particular regard to the number of obstructions and their coordinates. The algorithm developed is based upon the work of Edwards and Durkin, (1968) and Meeks, (1983).

Typical profiles for path loss estimation are between two and thirty kilometres, so once the profile has been interpolated from the DEM/DTM, the elevations need to be corrected for earth curvature. The individual propagation losses are then calculated for the free space loss (F), the reflection loss (R), the siting loss (S) and the diffraction losses (D_i, D_j, D_k) for up to three obstructions. The total path loss (T) is then calculated as : $T = F + R + S + D_i + D_j + D_k$. Field experiments have shown this algorithm to have a standard deviation of 6dB.

Results

The original grid DEM consists of points sampled at 50 metre intervals, which have been interpolated from the contours of the Ordnance Survey (O.S.) 1:50,000 scale Landranger Map Series. These heights have been rounded to the nearest metre. Data is supplied by the O.S. in 20x20 km cells, the southwest corner of which is an even-numbered 10 km National Grid (NG) value. The 401x401 pixel DEM used in this study is referenced as ST08. This cell covers typical terrain of the South Wales valleys (ie. hilly regions cut by steep, deep valleys). The elevation range is 451 metres (ie. minimum 19 m and maximum 470 m), with the mean elevation being 173.17 metres.

The models considered in this study included regular grid DEMs sampled or sub-sampled at 50m, 100m, 150m and 200m intervals (using two bytes of storage per elevation and one byte per elevation in a compressed form); triangulated irregular networks (for maximum surface error tolerances of 10m, 15m and 20m); surface patch quadrees (for maximum surface error tolerances of 10m, 15m and 20m); polynomial surface patches (for degrees three, four and five, each fitted to 10x10 pixels per patch); and microgrids (for 80x80 grids of 5x5 pixels using four two-byte words per grid for differences represented by two bits and three four-byte words per grid for three bit differences). For the surface patch quadtree, a limitation on the size of the modelled DEM (ie. $2^n \times 2^n$ pixels) is imposed. Therefore, the results below are taken as an average of a number of 256x256 pixel quadrees within the 401x401 pixel region of coverage.

One thousand profiles were randomly generated and applied to each of the models. (The path loss results for the surface patch quadtree are therefore not directly comparable with the other methods, due to its shorter distance range, but do give an indication of its performance related to the regular grid DEM).

The results of this study for the implemented models are shown in Table 1 for storage requirements, Table 2 for profile retrieval time, Table 3 for elevation errors and Table 4 for radio path loss errors.

Terrain Model	Percentage Storage
ST08 - 50 m Grid (2 Bytes)	100.000
TIN - Tolerance 10 mtrs	33.210
- Tolerance 15 mtrs	22.897
- Tolerance 20 mtrs	18.768
SPQ - Tolerance 10 mtrs	29.960
- Tolerance 15 mtrs	18.980
- Tolerance 20 mtrs	13.280
Poly - Degree 3	20.000
- Degree 4	30.000
- Degree 5	42.000
ST08 - 100m Grid (2 Bytes)	25.000
- 150m Grid (2 Bytes)	11.223
- 200m Grid (2 Bytes)	6.250
ST08 - 50 m Grid (1 Byte)	50.000
- 100m Grid (1 Byte)	12.500
- 150m Grid (1 Byte)	5.613
- 200m Grid (1 Byte)	3.125
MGrid - (2-Bit Differences)	16.000
- (3-Bit Differences)	24.000

Table 1 : Storage Requirements of Models as a Percentage of Original 50m 2-Byte Grid

Terrain Model	Profile Gen. Time	Time Ratio to 50m Grid
ST08 - 50 m Grid (2 Bytes)	0.1617	1 : 1.000
TIN - Tolerance 10 mtrs	16.733	1 : 103.48
- Tolerance 15 mtrs	13.342	1 : 82.511
- Tolerance 20 mtrs	11.181	1 : 75.331
SPQ - Tolerance 10 mtrs	180.330	1 : 1115.2
- Tolerance 15 mtrs	118.390	1 : 732.16
- Tolerance 20 mtrs	88.050	1 : 544.53
Poly - Degree 3	0.1620	1 : 1.002
- Degree 4	0.1629	1 : 1.007
- Degree 5	0.1635	1 : 1.011
ST08 - 100m Grid (2 Bytes)	0.0824	1 : 0.510
- 150m Grid (2 Bytes)	0.0555	1 : 0.344
- 200m Grid (2 Bytes)	0.0422	1 : 0.261
ST08 - 50 m Grid (1 Byte)	6.7332	1 : 41.640
- 100m Grid (1 Byte)	3.2869	1 : 20.327
- 150m Grid (1 Byte)	2.1657	1 : 13.393
- 200m Grid (1 Byte)	1.6286	1 : 10.071
MGrid - (2-Bit Differences)	15.3070	1 : 94.665
- (3-Bit Differences)	12.8470	1 : 79.447

Table 2 : Average Profile Interpolation Time in Millisecs & Time Ratio compared to Bilinear Interpolation in the Original 50m 2-Byte Grid (DEC VAX 8650 processor).

Terrain Model	% of Re-Interpolated Points within					Absolute Ave. Error	Maximum Error
	0-5m	5-10m	10-15m	15-20m	Ov. 20m		
ST08 - 50 m Grid (1/2 Bytes)	100.000	---	---	---	---	---	---
TIN - Tolerance 10 mtrs	86.534	13.467	---	---	---	2.3580	10.000
- Tolerance 15 mtrs	72.675	22.893	4.442	---	---	3.3391	15.000
- Tolerance 20 mtrs	63.495	26.250	8.635	1.615	---	4.8492	20.000
SPQ - Tolerance 10 mtrs	90.520	9.480	---	---	---	1.8663	10.000
- Tolerance 15 mtrs	78.624	18.121	3.255	---	---	2.9462	15.000
- Tolerance 20 mtrs	67.658	22.884	7.877	1.582	---	4.0806	20.000
Poly - Degree 3	90.885	7.893	1.012	0.171	0.040	2.0289	35.661
- Degree 4	95.579	4.100	0.289	0.026	0.006	1.5030	26.762
- Degree 5	97.875	2.104	0.101	0.009	0.001	1.1729	24.606
ST08 - 100m Grid (1/2 Bytes)	96.784	2.909	0.277	0.025	0.005	0.9955	30.000
- 150m Grid (1/2 Bytes)	90.583	7.708	1.381	0.263	0.066	1.9138	31.444
- 200m Grid (1/2 Bytes)	81.737	13.406	3.469	0.997	0.391	2.8654	38.000
MGrid - (2-Bit Differences)	90.841	7.276	1.362	0.374	0.147	2.0805	41.500
- (3-Bit Differences)	95.455	4.420	0.124	0.001	---	1.5563	17.600

Table 3 : Error Performance of Terrain Models compared to Original 50m Grid. (Absolute Average Error and Maximum Absolute Error given in Metres)

Terrain Model	f=200 MHz (1000 Profiles)			f=900 MHz (1000 Profiles)			f=1800 MHz (1000 Profiles)		
	Av.Loss	No<6dB	<12dB	Av.Loss	No<6dB	<12dB	Av.Loss	No<6dB	<12dB
ST08 - 50 m Grid (1/2 Bytes)	---	1000	---	---	1000	---	---	1000	---
TIN - Tolerance 10 mtrs	2.622	885	966	2.185	862	963	2.609	851	932
- Tolerance 15 mtrs	3.121	846	958	2.559	833	957	3.052	821	906
- Tolerance 20 mtrs	3.536	817	946	3.002	803	937	3.646	782	884
SPQ - Tolerance 10 mtrs	3.026	864	960	2.887	835	952	3.212	819	942
- Tolerance 15 mtrs	3.614	815	939	3.466	785	944	3.969	764	918
- Tolerance 20 mtrs	4.121	779	925	3.972	761	923	4.683	713	887
Poly - Degree 3	2.915	862	962	2.996	835	957	3.418	812	937
- Degree 4	2.484	890	969	2.451	876	966	2.784	857	952
- Degree 5	2.049	915	975	2.050	906	977	2.330	892	964
ST08 - 100m Grid (1/2 Bytes)	2.697	884	962	2.476	877	970	2.761	855	954
- 150m Grid (1/2 Bytes)	5.037	717	898	4.997	704	886	5.785	681	843
- 200m Grid (1/2 Bytes)	5.281	706	898	5.623	655	870	6.541	621	817
MGrid - (2-Bit Differences)	3.199	839	959	3.464	804	930	3.956	781	908
- (3-Bit Differences)	2.920	854	961	3.248	820	935	3.611	800	918

Table 4 : Average Radio Path Loss Error in decibels (dB) compared to Original 50m Grid for frequencies of 200MHz, 900MHz and 1800MHz, together with the number of profiles within 6dB and 12dB error (Out of 1000 Profiles).

Discussion

For the methods implemented, it is apparent that 'reasonable' approximations of the study terrain can be achieved with storage savings in excess of 70%. The methods considered can be differentiated according to whether or not they constrain the maximum error. The adaptive methods (TIN and surface-patch quadtree) have the potential to improve significantly on storage costs for more uniform terrain, whilst still constraining maximum error. Further investigations for O.S. grid DEM reference ST06 (which is less hilly than ST08), have shown that the storage reduction approaches 85% for both the TIN and surface patch quadtree at the 10m tolerance level.

Adaptive methods with controlled errors have a clear advantage over most other terrain approximation techniques. Maximum error in an unconstrained model cannot be pre-determined and is related to the variability of the terrain. However, unconstrained techniques can give a good overall performance in terms of absolute average error. The results from non-adaptive sub-sampled grids at the 100m interval highlight the redundancy in the original grid DEMs, but the arbitrary rejection of points at larger intervals can result in major errors. The use of TINs on the other hand, has shown that the terrain can be modelled to within a tolerance of ten metres using under 3.5% of the original grid points. This suggests the fact that storing irregularly sampled points (not necessarily as a TIN) has the potential to offer large storage savings.

For regular grid methods, error-free data compaction such as by Huffman codes, can always further improve the results. Thus storage overheads can be reduced significantly, with only a small increase in retrieval time. Preliminary results

have shown that the error-free Huffman coding gives an entropy of 2.4 bits per elevation for ST08. This represents an 85% storage saving for two-byte elevations or 70% for one-byte elevations, but a look-up table of codes and reference heights will also need to be stored. However, if this were applied to the 100m sub-grid of elevations, the storage saving would be in excess of 90% compared with the regular grid DEM, with no further increase in elevation or path loss error. If sub-sampling of the terrain is not suitable in certain regions, a variable grid data structure could be used. It should be noted that error-free data compaction techniques can also be applied to the other models. For example, the elevations stored in the quadtree can be compressed into one byte values, thus decreasing storage requirements by a further 40%.

The algorithms used for calculating the profile are not necessarily the fastest or most optimal. For example, with the surface patch quadtree, a better approach might be to trace the path of the profile from one end point to the other, so that a sorted list of interpolated points can be obtained directly. However, even given a different approach to profiling, the method will always require a significant degree of searching. This is also a critical factor for the TIN, since the intersected edges need to be followed through the network. The other models are all based on direct access of the data structure, which is significantly faster, especially if the data has not been compressed. Bit manipulations for decoding the data add significantly to retrieval time.

The results indicate that there is no outstanding method for estimating the radio path losses, but when storage costs are also taken into consideration, the 10m TIN, fourth order polynomial model and the 100m sub-sampled grid all give satisfactory results. The TIN, whilst not as accurate in terms of height errors as the surface patch quadtree at similar tolerance levels, gives superior results to it in terms of path loss. This indicates that the terrain features that will obstruct line-of-sight profiles, such as ridges have been incorporated in the TIN. The larger errors in polynomial models, which are usually in modelling the gradients of hills, rather than their peaks, have been shown to have a minor effect in the path loss algorithm. Similarly, the 100m sub-sampled grid does not lose too much in the way of important features.

Conclusions

Implementation of various terrain modelling methods has shown that it is possible to reduce the regular grid storage by 70%. However, this will usually incur some cost, such as increased elevation error and processing speed for retrieval operations. For the application of path loss estimation, the most promising approach is one that can directly access the data structure for retrieving the profile quickly. This suggests that grid-based approaches, including polynomial patches, offer the best solution to the problem.

Regarding the general problem of data storage, our results on a limited set of models have not provided compelling evidence of the superiority of any single modelling method. However, error-constrained adaptive techniques based on TINs and surface patch quadtrees appear to have the potential to achieve relatively higher savings for low variability terrain. It may also be remarked that our TIN data structure included a major overhead in pointer storage. This

is highlighted by the fact that the 10m error-constrained surface required only 3.5% of original grid points. It is possible to envisage storing only these triangle vertices and reassembling the triangulation at retrieval time. In any event the greatest space savings can be expected if error-free data compression techniques, such as Huffman coding are applied after error-constrained modelling. Our results indicate that this can result in space savings of the order of 90%.

References

- Boehm, B.W. (1967) *"Tabular Representation of Multivariate Functions - with Applications to Topographic Modelling"*, Memo. RM-4636-PR, Rand Corporation, California, Feb., 67 pages.
- Burrough, P.A. (1986) *Principles of Geographic Information Systems for Land Resources Assessment*, Oxford University Press, New York, 193 pages.
- Chen, Zi-Tan; Guevara, J.A. (1987) *"Systematic Selection of Very Important Points (VIP) from Digital Terrain Models for Constructing Triangulated Irregular Networks"*, Proc. of Auto-Carto VIII, March 29th-April 3rd, pp.50-56.
- Chen, Zi-Tan; Tobler, W. (1986) *"Quadtree Representations of Digital Terrain"*, Proc. of Auto-Carto London, Vol.1, Sept.14-19th, pp.475-484.
- Davis, J.C. (1986) *Statistics and Data Analysis in Geology*, John Wiley & Sons, 2nd Ed., 646 pages.
- DeFloriani, L.; Falcidieno, B.; Pienovi, C. (1983) *"A Delaunay-Based Method for Surface Approximation"*, Proc. of Eurographics '83, Univ. of Zagreb, Aug.31st-Sept.2nd, pp.333-350.
- Douglas, D.H. (1986) *"Experiments to Locate Ridges and Channels to Create a New Type of Digital Elevation Model"*, Cartographica, Vol.23, No.4, Dec., pp.29-61.
- Edwards, R.; Durkin, J. (1969) *"Computer Prediction of Service Areas for v.h.f. Mobile Radio Networks"*, Proc. of the IEEE, Vol.116, Electronics, No.9, Sept., pp.1493-1500.
- Fowler, R.J.; Little, J.J. (1979) *"Automatic Extraction of Irregular Network Digital Terrain Models"*, Computer Graphics, Vol.13, No.1, Jan., pp.199-207.
- Gargantini, I. (1982) *"An Effective Way to Represent Quadtrees"*, Comm. of the ACM, Vol.25, No.12, Dec., pp.905-910.
- Jenson, S.K. (1985) *"Automated Derivation of Hydrologic Basin Characteristics from Digital Elevation Model Data"*, Proc. of Auto-Carto VII, March 11th-14th, pp.301-310.
- Junkins, J.L.; Miller, J.R.; Jancaitis, J.R. (1973) *"A Weighting Function Approach to Modelling of Irregular Surfaces"*, Journal of Geophysical Research, Vol.78, No.11, pp.1794-1803.
- Krumbein, W.C.; Graybill, F.A. (1965) *An Introduction to Statistical Models in Geology*, International Series in the Earth Sciences, McGraw-Hill, 475 pages.
- Leifer, L.A.; Mark, D.M. (1987) *"Recursive Approximation of Topographic Data using Quadrees and Orthogonal Polynomials"*, Proc. of Auto-Carto VIII, March 29th-April 3rd, pp.650-659.
- McCullagh, M.J.; Ross, C.G. (1980) *"Delaunay Triangulation of a Random Data Set for Isarithmic Mapping"*, The Cartographic Journal, Vol.17, No.2, Dec., pp.93-99.
- Meeks, M.L. (1983) *"VHF Propagation over Hilly, Forested Terrain"*, Antennas and Propagation, Vol.3, pp.483-489.
- Petrie, G. (1987) *"Data Interpolation and Contouring Methods"*, Proc. of Terrain Modelling in Surveying & Civil Engineering, Univ. of Surrey, April 7th-9th / Univ. of Glasgow, Sept.1st- 3rd.
- Peucker, T.K.; Douglas, D.H. (1975) *"Detection of Surface-Specific Points by Local Parallel Processing of Discrete Terrain Elevation Data"*, Computer Graphics and Image Processing, Vol.4, pp.375-387.
- Peucker, T.K.; Fowler, R.J.; Little, J.J.; Mark, D.M. (1978) *"The Triangulated Irregular Network"*, Proc. of the DTM Symposium, St. Louis, Missouri, May 9th-11th, pp.516-540.
- Sibson, R. (1978) *"Locally Equiangular Triangulations"*, The Computer Journal, Vol.21, No.3, pp.243-245.
- Williams, C.M. (1986) *"The Geometric Modeling and Compression of Terrain Data"*, Proc. of 2nd Int. Conf. on Spatial Data Handling, Seattle, Washington, July 5-10th, pp.158-170.

IMPLICIT TRIANGULATIONS FOR LARGE TERRAIN DATABASES

David Kidner & Christopher Jones

Department of Computer Studies

The Polytechnic of Wales

Treforest, Mid Glamorgan, Wales, U.K. CF37 1DL

e-mail : DBKIDNER@UK.AC.POW.GENVAX

ABSTRACT

Triangulated digital terrain models (DTMs) hold several advantages over regular grid DTMs for surface representation, particularly with respect to reducing the volume of data while still incorporating original surveyed points. However, when stored explicitly, this data volume may still be very large, primarily due to pointer maintenance in defining the adjacency relationships of the triangulation. Current schemes, employing nodes, edges or triangles as the primary entities of the DTM, require the storage of adjacent elements to maintain TIN topology. In contrast, an alternative approach, termed implicit triangulation, involves storing only triangle vertices corresponding to critical points. The surface is then reconstructed locally in an area of interest, using the Delaunay triangulation. This significantly reduces storage overheads by eliminating the need for permanent representation of adjacency relationships. Assuming that a typical node valency is six, and the storage space of a pointer is one third of a vertex, the space requirements of a triangulation can be reduced by at least two thirds. Spatial search of large implicit TINs, to access windows of interest, can be facilitated by conventional range search data structures, based for example on a regular grid tessellation. This form of spatial addressing allows compression of the vertex coordinates, such that a storage saving of 90% can be attained, when compared to a conventional vertex-based TIN. Subsequent use of the Delaunay triangulation provides the necessary consistency of reconstruction to retrieve the local surface at a pre-specified tolerance. In addition, implicit triangulation provides a flexible and convenient means of representing DTMs at multiple resolutions. This paper presents a comparative evaluation of implicit and explicit triangulation and assesses the benefits of both methods over regular grid DTMs.

1. INTRODUCTION

Digital terrain modelling addresses the problem of characterising the Earth's surface by either numerical or mathematical representations of a finite set of terrain measurements. The nature of terrain data structures depends largely upon the degree to which they attempt to model reality and/or the intended application(s) of the user. These user-specific approaches have led to the creation of a variety of digital terrain models (DTMs), many of which have not been fully exploited for the application of geographical information systems (GIS). Overviews and comparisons of some of the most popular DTMs have been made by Mark (1975) and Peucker (1978).

The regular rectangular grid is the most commonly used DTM, due to its simplicity, implicit coordinates, application efficiency and widespread availability of data in this format. However, its inability to adapt to terrain variability and the likelihood of data redundancy has led to the development of alternative methods. One such DTM is the triangulated irregular network or TIN (Peucker et al, 1978), which utilises 'surface-specific' points (eg. peaks, pits, passes, ridges and channels) to form a network of triangular facets. The model adapts itself to the roughness of the terrain, with no data redundancy, since all incorporated data represent these critical points. Methods of constructing the planar triangulation have received widespread attention (Gold, 1979). For most applications, such as interpolation, a good triangulation produces triangles which are as equiangular as possible, thus avoiding elongated triangles. The Delaunay triangulation has become accepted as the best approach for the creation of a TIN, since it satisfies this requirement and produces a unique solution in one pass of the data. De Floriani (1987) reviews the literature with respect to both static and dynamic algorithms for constructing the Delaunay triangulation.

As terrain data becomes widely available, the use of large terrain databases is an increasingly important consideration in the design and integration of DTMs within GIS. However, when a large number of data points are available, a TIN joining all the data can be highly inefficient in storage and for search and retrieval operations (De Floriani, 1987). The physical form of the TIN data structure can vary quite significantly, differing in the necessary storage space per unit and the completeness of the topological definition of the structure (Peucker, 1978). The less redundant the storage, the more the topological relations have to be searched or derived algorithmically instead of retrieved directly. This choice of storage or computing efficiency in the design of the data structure arises in the design of DTMs in general.

With the increasing demands of GIS users, the need for accurate terrain data at finer resolutions and at national levels has become apparent. Such vast volumes of data, even when stored as TINs may exceed the storage limits of many systems. However, a variety of DTM applications may only require access to a subset of the full terrain model at any one time, such as in profiling, visibility analysis, earthwork calculations, communication network siting etc. For such applications, the need for an accurate, error-constrained DTM that incorporates original surveyed points may be critical (Kidner et al, 1990). Whilst the TIN meets these requirements, the representation of its topological relationships may often be too inefficient when storage is of primary concern. For some TINs the storage costs may be greater than that of a regular grid DTM, from which it may have been derived. Some current approaches to overcoming the data handling problem for large terrain databases have focused upon either multiple resolution TINs, in which a pyramid-structure is used to represent the surface at different levels of resolution, or alternatively hierarchical TINs, in which different parts of the surface are represented at different levels of resolution (Samet, 1990; De Floriani, 1987).

This paper proposes the use of an implicit TIN derived from a dense regular grid DTM, in which only those triangle vertices corresponding to the critical points are stored. This DTM assumes that a Delaunay triangulation procedure is an integral part of the TIN, since the topology is 'retrieved' or reconstructed by its application to the vertices. An initial implicit TIN is created by a triangulation of the most critical surface points, which are supplemented by the iterative introduction of points which constrain the TIN to the user-prescribed error tolerance. If the surface model is too large to reside in memory, the TIN could be segmented for this derivation process. Once the dynamic triangulation is complete, the topological relationships defining the TIN are discarded and any partitioned data sets are merged. At the application stage, efficient spatial search of the data can restrict the retrieval of points to discrete windows of interest. At this local level, the subsequent use of the Delaunay triangulation will regenerate the original TIN surface, whilst maintaining the integrity of the original TIN within this region. The implicit TIN is particularly preferable for subset operations of the DTM, such as profiling, since the full topology of the explicit TIN is not maintained in memory.

An important prerequisite of the implicit TIN is an efficient spatial search of vertices. This can be accomplished with a grid overlay or box-sort structure, such as that described by McCullagh & Ross (1980). The use of such a secondary data structure permits the x, y and z coordinates of each vertex to be compressed into two bytes of storage. Storage savings of at least 90% over a conventional vertex-based TIN, can therefore be achieved. This substantially increased storage efficiency is obtained at a cost of a reduction in computing efficiency at the application stage. However, the application of parallel processing techniques for the retrieval of TIN topology and interpolation algorithms can significantly improve search and retrieval time (Ware & Kidner, 1991). The uniqueness of the Delaunay triangulation facilitates such a parallel approach.

2. DATA STRUCTURES FOR TRIANGULATED IRREGULAR NETWORKS (TINs)

The structure of a triangulated irregular network can be represented in a number of different ways. Three main approaches have evolved in the generation of a subdivision of a surface into triangular facets. The primary entities of each represent the three primitive topological entities of a TIN - vertices, triangles and edges. A TIN data structure for encoding a triangulation can be thought of as the combination of these basic entities and a set of adjacency relations (De Floriani, 1987). Woo

(1985) illustrates by virtue of an arrow diagram the nine possible relations that can be defined between pairs of primitive entities (Figure 1), where each arrow denotes an ordered relation between a pair of entities.

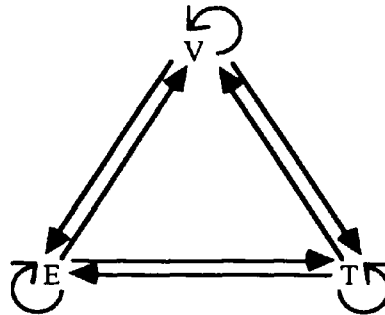
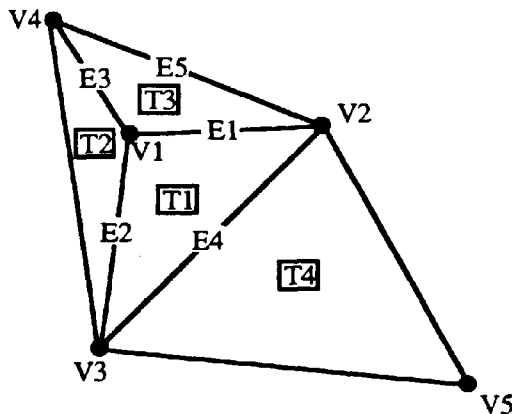


Figure 1 - Nine Possible Relations Between Pairs of Entities in a Triangular Grid.
(V:Vertices; E:Edges; T:Triangles).

De Floriani (1987) states that the topology of a triangular subdivision is completely and unambiguously represented by any suitably selected subset of these nine mutual adjacency relations. An illustration of these relations is presented below in Figure 2.



- 1 Vertex - Vertex : Given V1 Store V2, V3, V4
- 2 Vertex - Edge : Given V1 Store E1, E2, E3
- 3 Vertex - Triangle : Given V1 Store T1, T2, T3
- 4 Edge - Vertex : Given E1 Store V1, V2
- 5 Edge - Edge : Given E1 Store E4, E2, E5, E3
- 6 Edge - Triangle : Given E1 Store T1, T3
- 7 Triangle - Vertex : Given T1 Store V1, V2, V3
- 8 Triangle - Edge : Given T1 Store E1, E4, E2
- 9 Triangle - Triangle : Given T1 Store T2, T3, T4

Figure 2 - Illustration of the Nine Possible Relations Between Pairs of Entities .
in a Triangular Grid (where V_n :Vertices; E_n :Edges; T_n :Triangles).

Vertex-Based TIN

In all TIN data structures, the x, y and z coordinates of every surface-specific point or vertex need to be defined, together with an index which uniquely references each point. In vertex-based TINs, such points are considered the primary entities of the data structure. Associated with each vertex is a list of pointer values indicating the position (and number) of connected points or edges, emanating from that vertex. With respect to the adjacency relationships, only the vertex-vertex relations and two entities (vertices and edges) are stored. The vertex-edge relations can be easily implied, whilst the vertex-triangle relations can be located by examining adjacent vertices. Since each edge is stored twice, storage may be reduced considerably by representing each edge only once, but a global search is required to find all the topological relationships.

Triangle-Based TIN

The second type of data structure regards the triangles as the primary entities, although the coordinates of the vertices still need to be stored in a secondary file. Each triangle is uniquely referenced and is defined by pointers to three corner points of the vertex-file, together with pointers

to the three adjacent triangles. Thus, triangle-vertex and triangle-triangle relations are maintained within the data structure, whilst triangle-edge relations can be retrieved in constant time. This is the most popular form of representing a TIN, but it is not as storage-efficient as the vertex-based TIN. McKenna (1987) proposes the use of a hybrid data structure that utilises the list of connected points and the list of triangles, such that vertex-vertex and triangle-vertex relations are represented, whilst the vertex-edge and triangle-edge relations can be derived implicitly.

Edge-Based TIN

Heller (1990) advocates the use of an edge-oriented structure, since it is better suited for the swapping and splicing of triangle edges. De Floriani (1987) describes the modified winged-edge representation in which the three basic topological entities are stored together with the edge-vertex, edge-edge and edge-triangle relations. For each edge joining two vertices, its neighbouring triangular facets and two of its neighbouring edges are stored. Other relations may be derived efficiently, although the data structure can be considered to have the greatest storage overheads.

Each TIN data structure has evolved through the requirements of specific applications, such that they each have their own distinct advantages and disadvantages. The relationships that are incorporated within the TIN can be configured to the requirements of the application(s), such that a hybrid of topological relationships is allowable. However, the degree of TIN topology is directly related to storage costs. Factors which may further affect this depend upon the computer implementation of the data structure and whether the TIN supports static or dynamic triangulation. If a continuous update of the TIN is a prerequisite, such as for point insertion, the data structure will require sufficient flexibility to allow this. This may mean using linked or doubly-linked lists which place greater demands on storage space. Fixed-size lists or arrays can be more easily incorporated into static triangulation data structures, such that pointers are made redundant or can be replaced by indices to records. However, irrespective of these factors, it can be concluded that edge-based and triangle-based TINs are likely to require significantly more storage space than a vertex-based TIN.

From Euler's theorem, it can be shown that for a triangulation of N nodes, B of which are on the boundary, there are $2N-B-2$ triangles and a total of $3N-B-3$ distinct edges or $6N-2B-6$ pointers, if stored as edges from each vertex. For a vertex-based TIN derived from a regular grid DTM, the storage requirements may still be very high. For example, if 10% of original grid points are needed to represent the surface satisfactorily, then the storage costs of the TIN could approach that of the original grid DTM. If the x , y and z coordinates together with the index correspond to $4N$ and the pointers to the connected points (edges) equal $6N-2B-6$ values, the storage costs are approximately $10N$ or 100% of the original grid storage. This assumes that the coordinates can be represented in the same storage space as the original grid elevation values and coordinate values are equivalent in storage to pointer values. The triangle-based TIN will require even greater storage, since for each of the $2N-B-2$ triangles, the three vertices and the three neighbouring triangles are stored ($12N-6B-12$), together with the vertex coordinates ($3N$). This corresponding storage requirement is equivalent to $15N$ or 150%, if 10% of the original grid points are used.

Whilst a TIN may be comprised of only a small proportion of original grid points, storage of the necessary topology to represent the spatial adjacency relations may incur too high a penalty. This is particularly true for large terrain databases. The explicit representation of adjacency relationships, such as edges in a vertex-based TIN, creates duplication of topology or data redundancy within the model. This can be illustrated by the fact that the $3N-B-3$ edges are stored using $6N-2B-6$ pointers. However, there is no redundancy or duplication within the implicit TIN, since no explicit topology is maintained.

The implicit TIN can produce substantial storage savings compared to a conventional TIN and even greater savings compared to a regular grid DTM. Since most TINs have to perform searches for some topological relationships, the implicit TIN expands this search to derive all necessary topological relationships. Only the triangle vertices are stored, so all applications will require the derivation of edges or triangles as required. However, these implicit relationships are derived by the application of the Delaunay triangulation to the vertices. As such, the triangulation algorithm is an

integral part of the DTM, in the same way as a triangle-search algorithm is associated with a vertex-based TIN. Efficient spatial search of the vertices minimises retrieval time, since only the topology required within a specific window of interest needs to be derived.

3. TRIANGULATION ALGORITHM

Before a TIN can be constructed, the points or vertices defining the triangular network have to be derived. This is often accomplished by selecting surface-specific points from a dense regular grid DTM according to some criteria. The selection criteria used for the implicit TIN is that the critical terrain features are supplemented by elevations that constrain the maximum error within the TIN to some predetermined tolerance. There are many algorithms that can be used to select these surface-specific points and lines, including those of Peucker & Douglas (1975), Douglas (1986) and Chen & Guevara (1987). However, the relative importance and contribution that each point makes in a new surface model, such as a TIN, cannot be determined to such an extent that average or maximum error can be forecast. Thus there is no ideal method (requiring only one or two passes of the data) which detects critical points or lines, eliminates superfluous data and supplements critical points that are missing. In practice, an iterative procedure can be applied, whereby points are successively selected until a pre-defined error tolerance is satisfied. Such approaches define an initial set of 'very' critical points, (using some of the algorithms described above), which when triangulated are supplemented by the insertion of the unused point of largest error (Fowler & Little, 1979; De Floriani et al, 1983). Heller (1990) also describes an algorithm to accomplish this, together with further consideration of some of the implementation issues.

The construction of the implicit TIN uses a sparse set of critical points derived from a grid DTM using the algorithms of Peucker & Douglas (1975) and Chen & Guevara (1987). The choice of initial critical points is arbitrary to some extent, but these algorithms allow thresholding tolerances to adjust this selection process to individual preferences. A Delaunay triangulation of these initial points is then constructed using the criteria set out in the algorithm of McCullagh & Ross (1980). Lee & Schachter (1980) discuss in greater detail the geometric properties of the Delaunay triangulation and also present two algorithms for its construction. The original grid DTM is interpolated from this TIN such that the errors associated with all unused points can be determined. The point with the largest error or priority is then inserted into the TIN using a local re-triangulation and the list of priorities is updated accordingly. This continues until the TIN satisfies the error constraints imposed upon it by the user. The explicit data structure used for the creation of the TIN must support the flexibility required for dynamic triangulation. However, as the topology will be discarded after its creation, the choice of data structure is arbitrary, although Heller (1990) recommends an edge-based or 'dual-edge' structure for such operations. An advantage of dynamically selecting TIN vertices is that the surface can be represented at different levels of resolution or error-constraints in a similar manner to the Delaunay pyramid of De Floriani (1989), but without explicit triangle descriptions. For example, an initial TIN at the top level may be supplemented by sets of points at lower levels which constrain the triangulation to maximum errors at say, 10 or 5 metre intervals. For the implicit TIN, this presents an easier form for representation, since only the vertices of the TIN are stored at any level, together with the spatial index for searching.

4. SPATIAL SEARCH FOR TRIANGULATION RECONSTRUCTION

An important characteristic of the Delaunay triangulation is its uniqueness for a given set of points. When triangulating a subset of the database of vertices however, the triangulation at the boundary of the subset cannot be guaranteed to be equivalent to the original global triangulation, since neighbouring vertices outside the region of interest will not be taken into account. To ensure that this does not have any major effect upon the required application, the spatial search of vertices for triangulation should allow a 'reasonable' margin of points around the application window. This margin should be dependent upon the sparseness of the data within the terrain model. For any spatial window, surround conditions can be set such that the search for vertices extends outwards until a minimum of points is obtained which ensures the accuracy of the reconstruction will not be

affected by the boundary.

Spatial addressing of points is accomplished with a simple grid overlay or box-sort structure (McCullagh & Ross, 1980), which allows fast access to the implicit TIN vertices. It assumes that the points have been pre-sorted in both the x and y dimensions. However, if the vertices are derived from a regular grid DTM, the data can be written directly to a file in this 'sorted' order. Each box or grid cell maintains a pointer indicating the position of the first point within the list of vertices, such that any number of points may reside in a box. Therefore, any point of known coordinates can be directly mapped into a box address, such that the vertices within that cell can be directly accessed from the vertex file. Search operations are augmented by a 'radius of search' parameter, which determines the range of neighbouring cells to be accessed. The size of the grid overlay will determine the optimality of searching operations. McCullagh & Ross (1980) suggest a grid which allows an average of four points per cell. An even denser grid may reduce search time further, but at a cost of increased storage for the pointers. For the implicit TIN it was found that a good compromise between search time and storage was three to five points per grid cell. However, further consideration is needed for multiple resolution implicit TINs, which maintain multiple vertex files and grid overlays. If the density of the grid is maintained at each level, there will be pointer redundancy due to many empty cells, especially if there are fewer and fewer vertices at each resolution. A compromise can be reached by decreasing the density of the grid with each increase in TIN resolution.

The properties of a grid cell addressing scheme can be utilised to compress the x, y and z coordinates into two bytes of storage. This assumes that the vertices are derived from a regular grid DTM. Since the grid provides a direct mapping of coordinates into cell addresses and vice versa, the coordinates of vertices within these cells can be represented at a local level. Such an addressing scheme would relate the local coordinates to a fixed point within the grid cell, such as a corner. The feasibility of this approach is dependent upon the resolution of the grid overlay and the average number of points per cell. In our application, the local coordinates of each x and y coordinate may be represented by three or four bits, with the elevation represented in eight to ten bits. In the example below (Figure 3), the grid overlay corresponds to 8x8 pixels, such that each x and y coordinate can be represented in three bits of storage. These values can be concatenated with each 10-bit elevation into a two-byte value, thus presenting a compact form of external file storage.

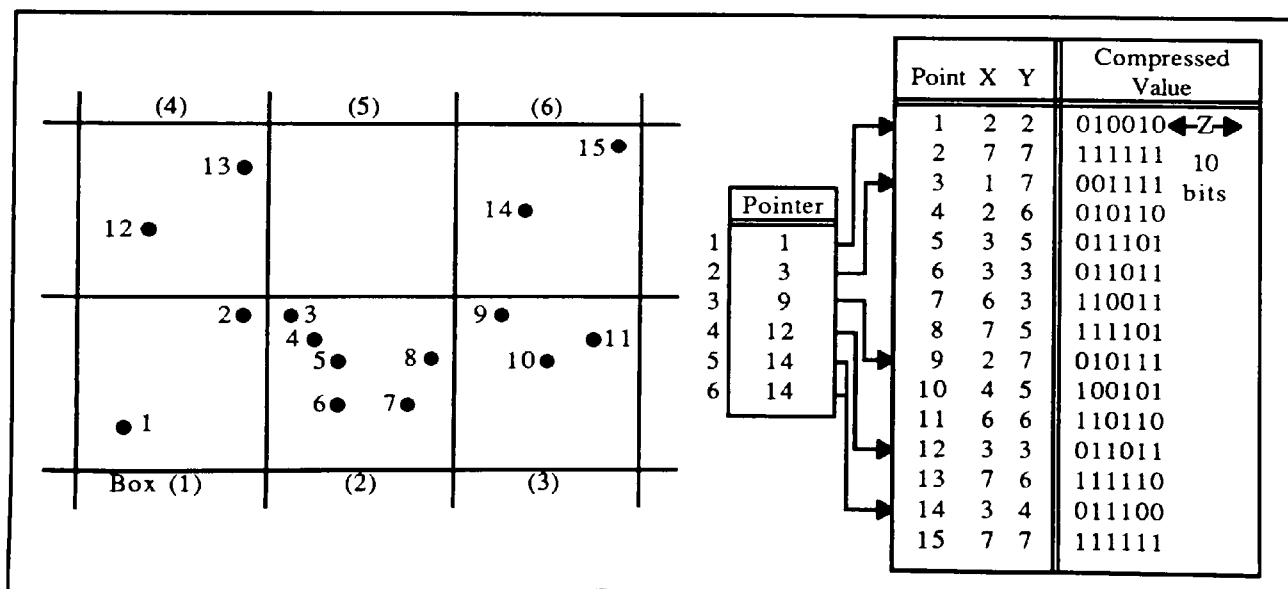
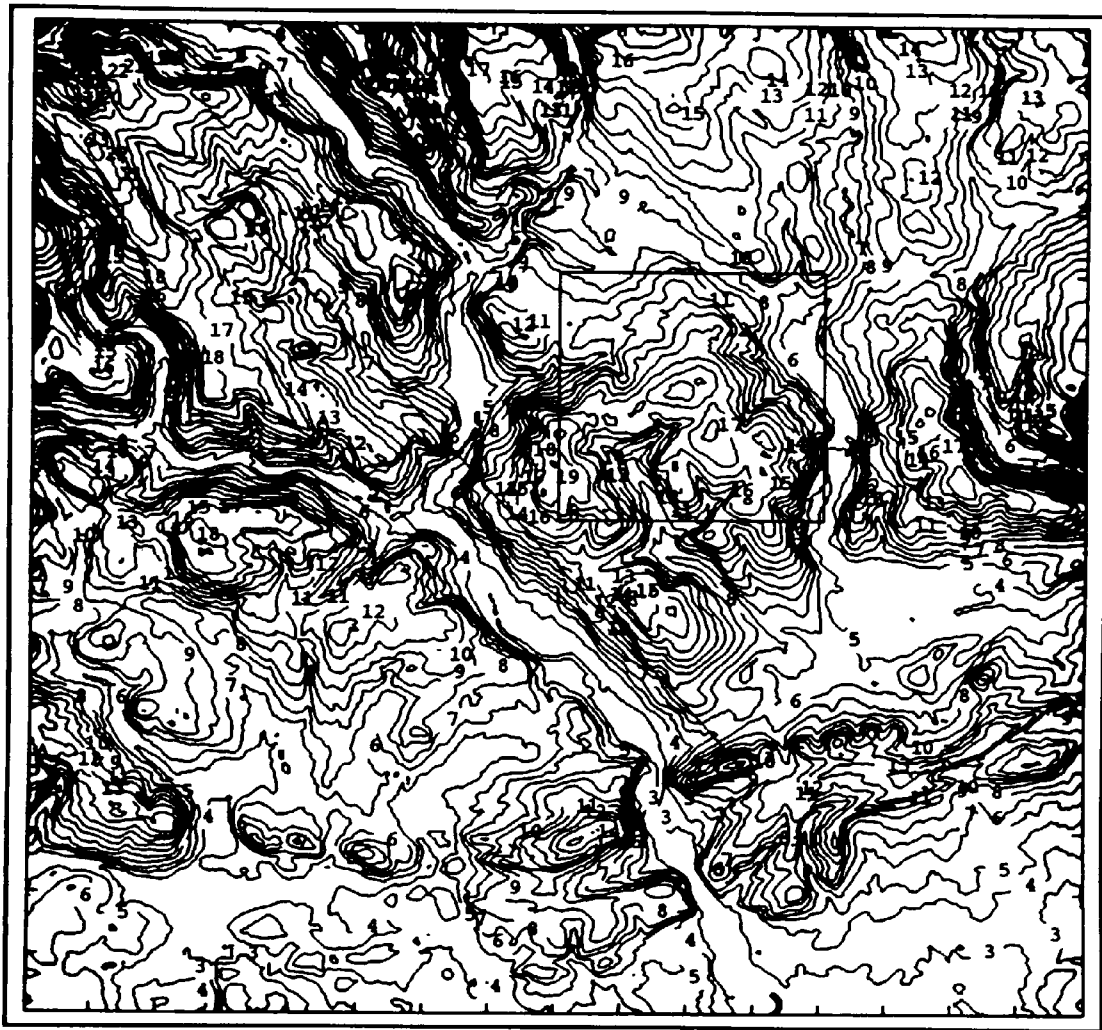


Figure 3 - Illustration of Coordinate Compression of Vertices in Local Coordinates.

The test data set used in the examples below, represents a 20 x 20 kilometre DTM of the South Wales valleys (National Grid Reference ST08/09/18/19 in Figure 4, below). The original grid DTM represents 401x401 nodes at 50 metre intervals. An implicit TIN was derived at three resolutions (15,

10 and 5 metres), the former of which is illustrated in Figure 5.



*Figure 4 - Contour Map of the Original 20 km x 20 km Test Data Set for South Wales.
(Contours at 20 metre Intervals, where Key Label 1 to 25 = 0 to 480 metres).*

Examples of the application of the implicit TIN can be illustrated for the two subset operations illustrated in Figure 4. The first consists of a square 5x5 kilometre region of interest, retrieved for the purpose of producing a contour map. For such an application, the grid cells covering the region of interest together with a 'sufficient' external boundary can be directly accessed to determine all the vertices of the local TIN. Once the convex hull has been determined, the implicit topology of the TIN is retrieved by the application of the Delaunay triangulation algorithm. This is illustrated in Figure 6 at resolutions of 15, 10 and 5 metres maximum absolute error. The first of these local TINs corresponds to the TIN of Figure 5. To retrieve a TIN at different resolutions, the vertex files may be concatenated up to the required resolution before triangulation. However, as in the case of Figure 6, if the grid overlay is at the same resolution throughout, or a multiple of the densest grid, the points within each can be directly accessed from the separate vertex files. Once the TIN topology has been reconstructed, the necessary interpolation for the required application can be performed locally. Figure 7 illustrates an example for the application of profiling. For the cross-section of length 14.82 kilometres shown on Figure 4, local implicit TINs were reconstructed at resolutions of 15, 10 and 5 metres maximum absolute error, respectively. In these examples, the spatial search of TIN vertices follows a systematic pattern. However, if a sparser TIN at a lower accuracy was required, the search for TIN vertices on each grid row or column may have to be extended, such that the application has sufficient points to form a convex hull which does not 'corrupt' the region of interest.

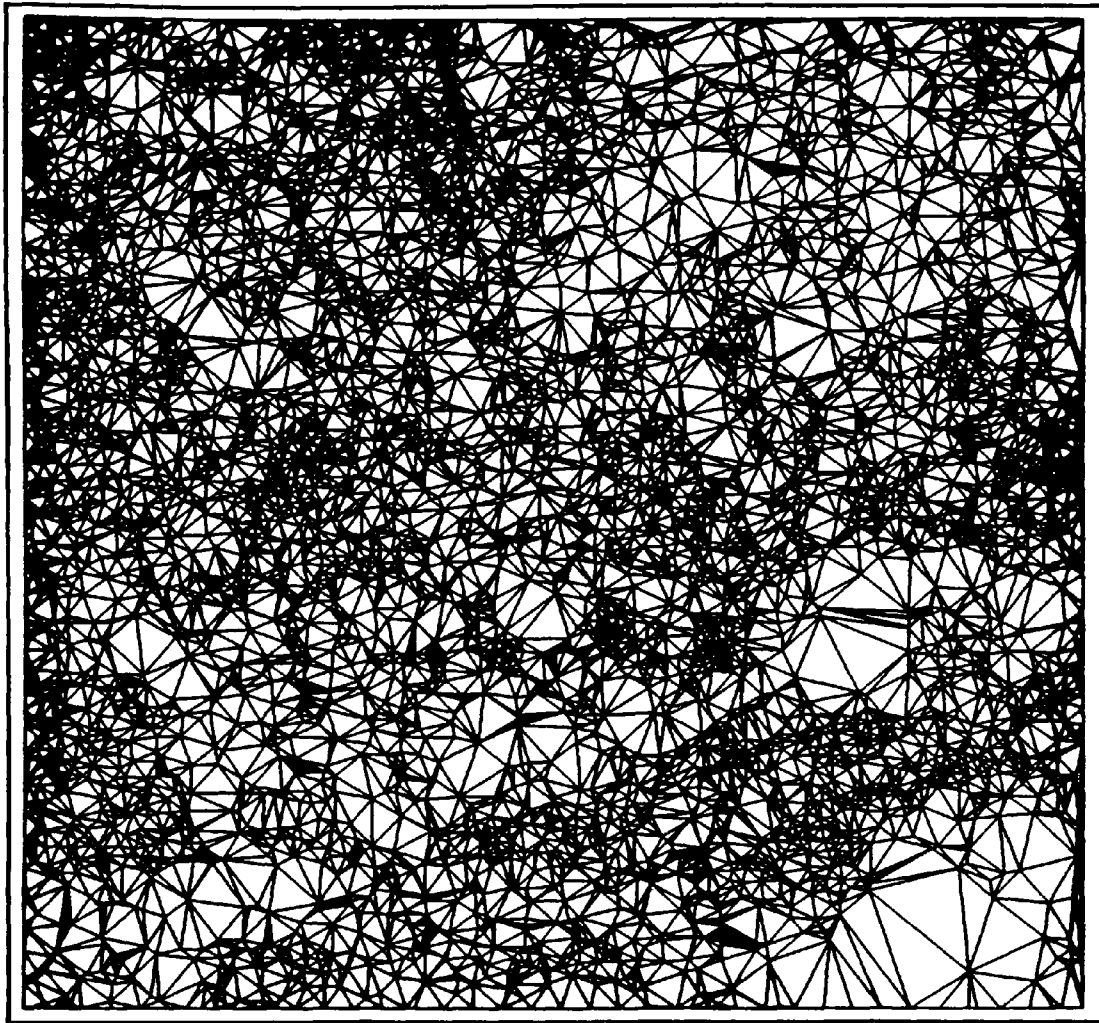


Figure 5 - Triangulated Irregular Network for Test Data Set at Accuracy Resolution of 15 metres.
(8278 Vertices, 309 Boundary Points, 16245 Triangles and 24522 Edges).

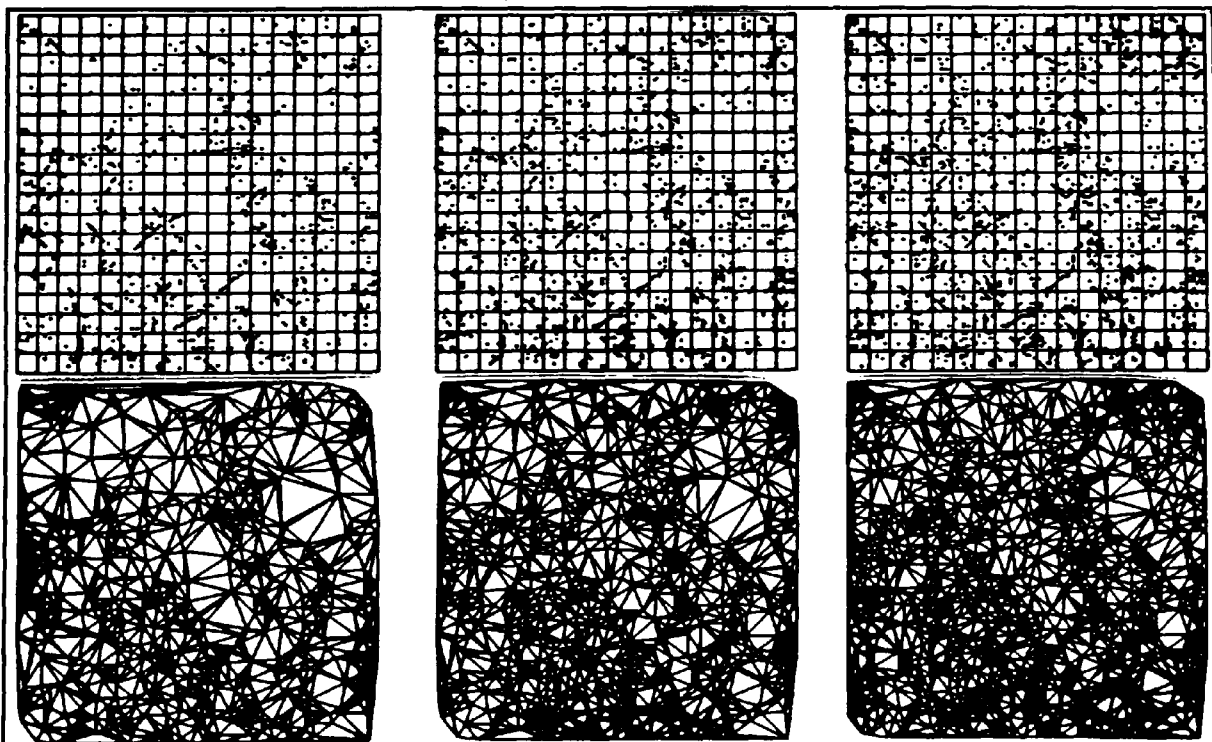


Figure 6 - Multiple Spatial Search of TIN Vertices for a Subset of the Original Surface (see Figure 4).

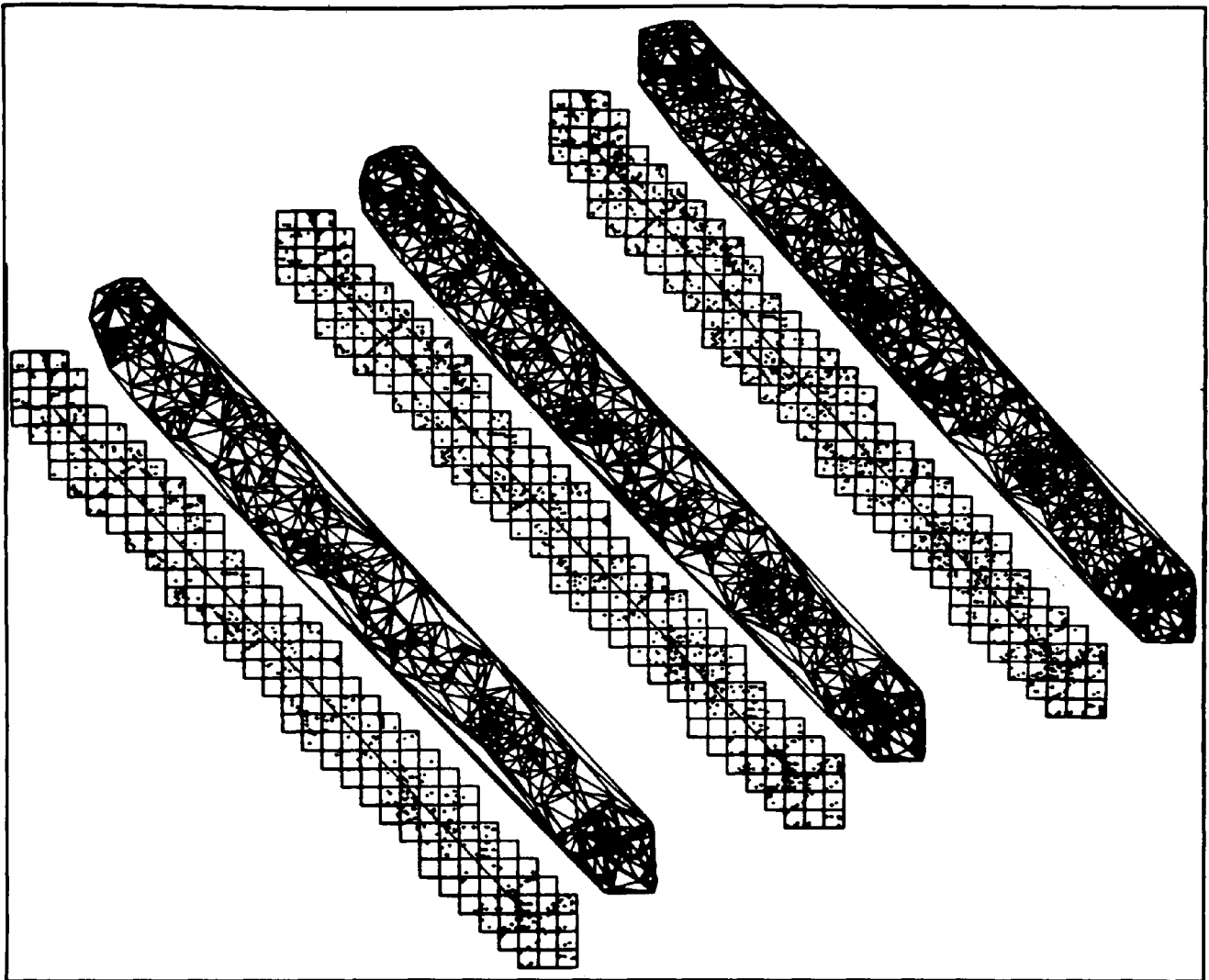


Figure 7 - Multiple Spatial Search of TIN Vertices for a Profile of the Original Surface(see Figure 4).

5. CONCLUSIONS

Conventional triangulated irregular networks incur a storage overhead due to the fact that they represent the topology of the triangulation explicitly. The storage space required for the topological data may be as much as two thirds of the total for the triangulation. This storage overhead can be avoided in a database by adopting a procedural approach to data retrieval in which a Delaunay triangulation algorithm is applied to stored vertices to replicate an originally triangulated surface. By adopting an iterative, error-constrained triangulation algorithm, the surface may be reconstructed locally to a user-specified resolution. This approach to the management of triangulated surfaces, termed implicit triangulation, is accompanied by the use of a spatial index which facilitates rapid access to spatial windows of interest. A simple grid-based scheme has been implemented in which vertex coordinates are stored relative to their containing grid cell. This provides additional storage savings, such that the storage of a conventional vertex-based TIN can be reduced by 90%. Response times for the reconstruction of triangulated surfaces may be minimised by implementing the triangulation procedures on parallel hardware. This is the subject of a related research project, the preliminary results of which are reported elsewhere (Ware & Kidner, 1991).

6. REFERENCES

- Chen, Zi-Tan, Guevara, J.A. (1987) "Systematic Selection of Very Important Points (VIP) From Digital Terrain Models for Constructing Triangulated Irregular Networks", Proc. of Auto Carto VIII, Mar.29-Apr.3, pp.50-56.
- De Floriani, L., Falcidieno, B., Pienovi, C. (1983) "A Delaunay-Based Method for Surface Approximation", Proc. of Eurographics '83, Zagreb, Aug.31-Sep.2, pp.333-350.
- De Floriani, L. (1987) "Surface Representations Based on Triangular Grids", The Visual Computer, Vol.3, No.1, Feb., pp.27-50.
- De Floriani, L. (1989) "A Pyramidal Data Structure for Triangle-Based Surface Description" IEEE Computer Graphics & Applications, Vol.9, March, pp.67-78.
- Douglas, D.H. (1986) "Experiments to Locate Ridges and Channels to Create a New Type of Digital Elevation Model", Cartographica, Vol.23, No.4, Dec., pp.29-61.
- Fowler, R.J., Little, J.J. (1979) "Automatic Extraction of Irregular Network Digital Terrain Models", Computer Graphics, Vol.13, No.1, Jan., pp.199-207.
- Gold, C.M. (1979) "Triangulation-Based Terrain Modelling - Where Are We Now?", Proc. of Auto Carto IV, Vol.2, Nov. 4-8, pp.104-111.
- Heller, M. (1990) "Triangulation Algorithms for Adaptive Terrain Modeling", Proc. of 4th International Symposium on Spatial Data Handling, Zurich, Vol.1, July 23-27, pp.163-174.
- Kidner, D.B., Jones, C.B., Knight, D.G., Smith, D.H. (1990) "Digital Terrain Models for Radio Path Profiles", Proc. of 4th International Symposium on Spatial Data Handling, Zurich, Vol.1, July 23-27, pp.240-249.
- Lee, D.T., Schachter (1980) "Two Algorithms For Constructing a Delaunay Triangulation", Int. Journal of Computer and Information Sciences, Vol.9, No.3, pp.219-242.
- McCullagh, M.J., Ross, C.G. (1980) "Delaunay Triangulation of a Random Data Set for Isarithmic Mapping", The Cartographic Journal, Vol.17, No.2, Dec., pp.93-99.
- McKenna, D.G. (1987) "The Inward Spiral Method : An Improved TIN Generation Technique and Data Structure for Land Planning Applications", Proc. of Auto Carto VIII, Mar.29-Apr.3, pp.670-679.
- Peucker, T.K., Douglas, D.H. (1975) "Detection of Surface-Specific Points by Local Parallel Processing of Discrete Terrain Elevation Data", Computer Graphics and Image Processing, Vol.4, pp.375-387.
- Peucker, T.K. (1978) "Data Structures for Digital Terrain Modules : Discussion and Comparison", In : G.Dutton (Ed.), Harvard Papers on Geographic Information Systems, First International Advanced Study Symposium on Topological Data Structures for Geographic Information Systems, Vol.5 - Data Structures : Surficial and Multi-Dimensional, Oct.16-21, 1977, 15 pages.
- Peucker, T.K., Fowler, R.J., Little, J.J., Mark, D.M. (1978) "The Triangulated Irregular Network", Proc. of the Digital Terrain Models (DTM) Symposium, May 9-11, pp.516-540.
- Ware, J.A., Kidner, D.B. (1991) "Parallel Implementation of the Delaunay Triangulation Within a Transputer Environment", Proc. of the Second European Conference on Geographical Information Systems.
- Woo, T.C. (1985) "A Combinatorial Analysis of Boundary Data Structure Schemata", IEEE Computer Graphics & Applications, Vol.5, No.3, pp.19-27.

ACKNOWLEDGEMENTS

This work has been partially funded by the Royal Signals and Radar Establishment (R.S.R.E.), Ministry of Defence, Procurement Executive, Malvern, Worcestershire, England.

PARALLEL IMPLEMENTATION OF THE DELAUNAY TRIANGULATION WITHIN A TRANSPUTER ENVIRONMENT

Andrew Ware⁺ & David Kidner^{*}

⁺Department of Mathematics and Computing

^{*}Department of Computer Studies

The Polytechnic of Wales

Treforest, Mid Glamorgan, Wales, U.K. CF37 1DL

e-mail : JAWARE or DBKIDNER@UK.AC.POW.GENVAX

ABSTRACT

Digital terrain modelling by triangulated irregular networks (TINs) is adaptive to surface variability, as only critical features are incorporated within the data structure. One such method for constructing a TIN is the Delaunay triangulation, an efficient technique which produces unique solutions using any random arrangement of points. As TINs are computationally expensive to construct from large numbers of points, recent research has focused on the use of optimal parallel algorithms in their implementation. The unique characteristics of the Delaunay triangulation lend themselves to parallel implementation on a transputer network. A transputer is a microcomputer with its own local memory which when networked to other transputers enables algorithms to run concurrently on a number of processors. This paper outlines the application of an array of transputers using algorithms encoded in OCCAM to implement a parallel version of the Delaunay triangulation. Major considerations of hardware and software implementation include the relationship between number of transputers used and algorithm efficiency, and a comparison between parallel and sequential algorithms. Experimental results support the conclusion that use of parallel algorithms running in a transputer environment, greatly enhances the computational efficiency of triangulation.

1. INTRODUCTION

Digital terrain modelling addresses the problem of characterising the Earth's surface by either numerical or mathematical representations of a finite set of terrain measurements. The regular rectangular grid is the most commonly used DTM, due to its simplicity, implicit coordinates, application efficiency and the widespread availability of data in this format. However, their inability to adapt to terrain variability and the likelihood of data redundancy has led to the development of alternative methods. One such DTM is the triangulated irregular network or TIN (Peucker et al, 1978), which utilises 'surface-specific' points (eg. peaks, pits, passes, ridges and channels) to form a network of triangular facets.

Many different triangulation algorithms have been defined for the surface representation of an arbitrary data set, in an attempt to satisfy a number of criteria used to determine a 'good' triangulation (Gold, 1979). McCullagh (1987) states that a triangulation should have the properties of stability, equilateralness and non-intersection for some applications, such as contouring, where an arbitrary triangulation may not be acceptable. The Delaunay triangulation meets these requirements and has been extensively used as a basis for surface modelling (De Floriani, 1987).

As DTMs are more widely used within GIS, the need has arisen for national data bases of terrain at fine resolutions. As a result, DTM and GIS data structures and their associated algorithms must evolve to handle the vast amounts of data that will become available. However, for a large number of data points, a TIN can be highly inefficient in storage and for search and retrieval operations (De Floriani, 1987). Hence the advantages of the TIN are in danger of being compromised by extensive data volumes. Whilst the surface-specific points are adaptable to surface roughness with no 'data redundancy', the representation of the TIN topology may incur extensive storage overheads (Kidner & Jones, 1991).

The problem of database storage overheads for a TIN can be alleviated by storing only the vertices from which the triangulation topology can be regenerated at the application stage using an appropriate algorithm (Kidner & Jones, 1991). However, for this and the generation of other static TINs, the triangulation process can be computationally expensive. Hence, the application of parallel processing for triangulating a set of points has recently received more widespread attention (Merks, 1986; El Gindy, 1986, 1990). The Delaunay triangulation is particularly amenable to a parallel implementation, due to its uniqueness characteristic (ie. for any set of points, the triangulation is singular). This paper details a prototype parallel implementation of the Delaunay triangulation and addresses some of the hardware and software considerations.

2. THE DELAUNAY TRIANGULATION

The Delaunay triangulation of a set of points is considered the 'best', since the triangles formed are as equiangular as possible and the longest sides of the triangles are as short as possible, thus avoiding thin and elongated facets. The resulting triangulation is unique, such that irrespective of starting point, the network of triangles formed will always be the same. This triangulation is the dual of the net of Thiessen polygons (Voronoi diagram, Dirichlet tessellation or Wigner-Seitz cells), in which any location on the plane is assigned to the polygon containing the nearest data point (Gold, 1979). Three lemmas can be distinguished which globally and locally define a Delaunay triangulation (Lee & Schachter, 1980) :

Lemma 1 : For any triangulation of N nodes, B of which are on the boundary (convex hull), there are $2N-B-2$ triangles and a total of $3N-B-3$ edges (Euler's Theorem).

Lemma 2 : Two vertices form a Delaunay edge, if and only if there exists a circle passing through the vertices that does not contain any other vertex.

Lemma 3 : Three vertices form a Delaunay triangle if and only if its circumcircle does not contain any other point in its interior.

From Lemma 2 and 3, a simple algorithm can be defined for the construction of the Delaunay triangulation, in which the properties of Lemma 1 are implicitly incorporated. The algorithm used in this study is based upon that of McCullagh & Ross (1980). The search of the neighbours of a vertex (Thiessen neighbours) proceeds in a clockwise direction around that point. For any known neighbour, the next neighbour is located as the vertex for which a circumcircle passes through the three points, with no other point inside the circle (Lemma 3). This is accomplished with a search circle passing through the vertices of the known edge, checking for an inscribed point in a clockwise direction. If no points are found, the size of the circle is increased, whilst if more than one point is located, the Thiessen neighbour which has the largest angle subtended from the known edge is selected.

This algorithm is satisfactory for 'arbitrary' located data, but since more and more TIN data are derived from regular grid DTMs, the degree of arbitrariness may be insufficient for a consistent triangulation. The regular coordinates of the vertices may cause degeneracies to occur, such that four or more points may lie on a circumscribing circle, each subtending the same angle from the known edge. This is equivalent to four or more Thiessen polygons meeting at one point. An example of this occurs for the four vertices of a regular grid square. By the above definitions, both diagonals are valid edges, thus causing an overlapping or intersection of triangles. If the edges are calculated from every vertex, the algorithm will produce a degenerate triangulation. McCullagh & Ross (1980) suggest selecting the point closest to the known neighbour providing it is not also closest to the other vertex of the known edge. However, this is not sufficient for a degeneracy of the above type. Instead a local decision rule is required which takes into account the nature of the surrounding points, since an arbitrary choice of edge may cause large elevation errors. For example, the diagonals of the four points forming a grid square could represent either a ridge or a channel, but the true feature can only be determined from examining the local terrain.

For a parallel implementation, there will be several initial points (one per processor), from which

the local triangulations are calculated. It is thus essential that the algorithm employed adheres to the criteria set out by McCullagh (1987), namely that of stability, equilateralness and non-intersection of triangles. This will require the triangulation to be unique for any starting point and is the most important implementation consideration. McCullagh & Ross (1980) have shown, for the Delaunay triangulation, that the resulting network of triangles is unique, irrespective of starting point.

3. PARALLEL PROCESSING IN A TRANSPUTER ENVIRONMENT

The procedure for Delaunay triangulation may be considered rather convoluted and thus slow in execution. This is primarily due to the search time required to locate the Delaunay edges or Thiessen neighbours of each point (see Section 4). In order for the triangulation to be executed in real time, a system has been developed to facilitate implementation using parallel processing techniques. Parallel processing involves either splitting the application or process to be performed into several sub-processes and performing these on different processors concurrently, or splitting the data that is to be processed between a number of processors and executing multiple copies of the process simultaneously. It should be noted that some algorithms are inherently sequential and, even for those that are not, it is often a non-trivial task to adapt sequential algorithms for efficient execution on parallel machines. Ware et al (1990) found converting from sequential to parallel processing may necessitate a completely new approach, since a simple transformation of the sequential code may prove impossible. Conversely there are occasions when inherent parallelism in the problem can give excellent results with only minor changes to the sequential algorithm.

The processor chosen for the parallel implementation of the Delaunay triangulation was the transputer¹. There are several chips in the transputer family, but the two most important ones are the T414 and T800, both of which rate 10 MIPS (million instructions per second) at 20 MHz. The T414 has 2K of on board RAM, whilst the more powerful T800 has 4K, as well as its own on-board floating point unit, which works in parallel with the main CPU and is capable of 1.5 Mflops (million floating point operations per second). In the prototype system described below, the host transputer is a T800, whilst the network transputers are T414s. Overall performance would be substantially increased if all the network transputers were substituted for T800s.

Transputers can be linked together to form a network of processors such that the workload of the application can be distributed across it. Not only does such a system offer a great deal of computing power, but its performance can be extended as required, by adding more transputers to the network. The network transputers are available as a fixed number per printed circuit board (PCB). For the prototype systems discussed in this paper, there are four transputers per PCB. Algorithms that are to be executed on the network may be implemented in a number of high level computer languages. However, if maximum efficiency is to be achieved they should be written in OCCAM, a programming language designed specifically for the transputer.

The initial prototype system was implemented on a network of four transputers arranged to form a bidirectional loop (Figure 1). The network is connected to a host transputer which is responsible for reading the data from disk before packaging it to be sent to the network. The whole system is housed in an IBM PC and has access to the host computer disk and input/output system.

To enable the OCCAM-encoded processes to run concurrently on several processors, a means of sending data from processor to processor is required. To keep communication overhead to a minimum it is important that this facility sends data via the best possible route around the network and that the number of times a processor needs to communicate with other processors is optimised. It is also important to ensure that processor power is used as effectively and efficiently as possible. To help meet these criteria the following rules were developed :

- 1) The processors should be kept 'busy' as soon as possible and for as long as possible.

¹ The word transputer is a composite of **transistor** and **computer**. The name reflects its design, in that it consists of a processor, memory and communications facilities built onto a single chip.

- 2) Data transfer between processors should be via the best possible route around the network. This is not necessarily the shortest as 'traffic jams' might develop along certain paths.
- 3) Data transfer should be kept at an optimum level to achieve maximum efficiency, such that when transfer is required it should be given priority over other tasks (this enables the receiving transputers to make use of the transferred information as soon as possible).

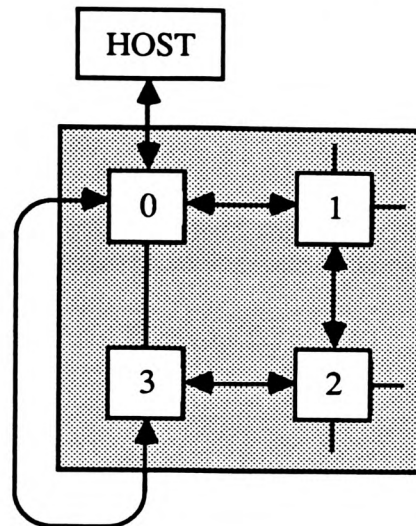


Figure 1 - Prototype System of Four Transputers.

(N.B. The flexible link between transputers 0 and 3 allows future expansion of the network).

The optimum level of information transfer depends on two factors. First, it may be quicker to duplicate processes on more than one processor, than to send information from processor to processor. Second, if one processor is generating information to be shared by other processors in the network, then the information can be passed at one of three stages. These stages being :

- (i) after all the information has been generated,
- (ii) after a given amount of information has been generated,
- (iii) as and when the information is generated.

Whilst this option depends on the system being implemented, the correct choice of approach for information passing is crucial to the efficient execution of parallel processes. When making this decision, the system designer should bear in mind rule (1). Timings have shown, that in the prototype system described, option (ii) proved to be the best choice.

To facilitate communication between the transputers in the network, each transputer has, in addition to the main algorithm being executed, a set of 'communication procedures'. These procedures allow the processor to receive and send data in the required manner. Data to be sent from one processor to another are packaged into a one-dimensional array. The first element indicates the destination address, the second indicates the source address, whilst the third represents the quantity of data being sent. The sending processor uses the first two pieces of information to determine the best route to the required destination. This communication packet is depicted below in Figure 2.

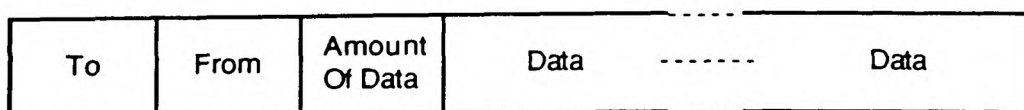


Figure 2 - Format For Passing Data Around the Network.

Each transputer in the network therefore has the following processes executing concurrently :

- (a) A 'get info' process to receive data from both the clockwise and anticlockwise transputers.
- (b) A multiplexer (Multi-Clock) to send data to the next clockwise transputer. This data may have been processed by the sending transputer, or the sending transputer may be acting as a link in the chain. (The multiplexer is required to collect data from the 'get info' and 'process request' routines and pass it on via a single connection).
- (c) A multiplexer (Multi-AClock) to send data to the next anticlockwise transputer. Again this data may have been processed by the sending transputer, or the sending transputer may be acting as a link in the chain.

In addition, the first transputer in the network has the facility to communicate with the host transputer, via the Multi-Host multiplexer. Figure 3 shows the processes that run concurrently on the first transputer, while the other transputers in the network have all but the Multi-Host process running concurrently on them.

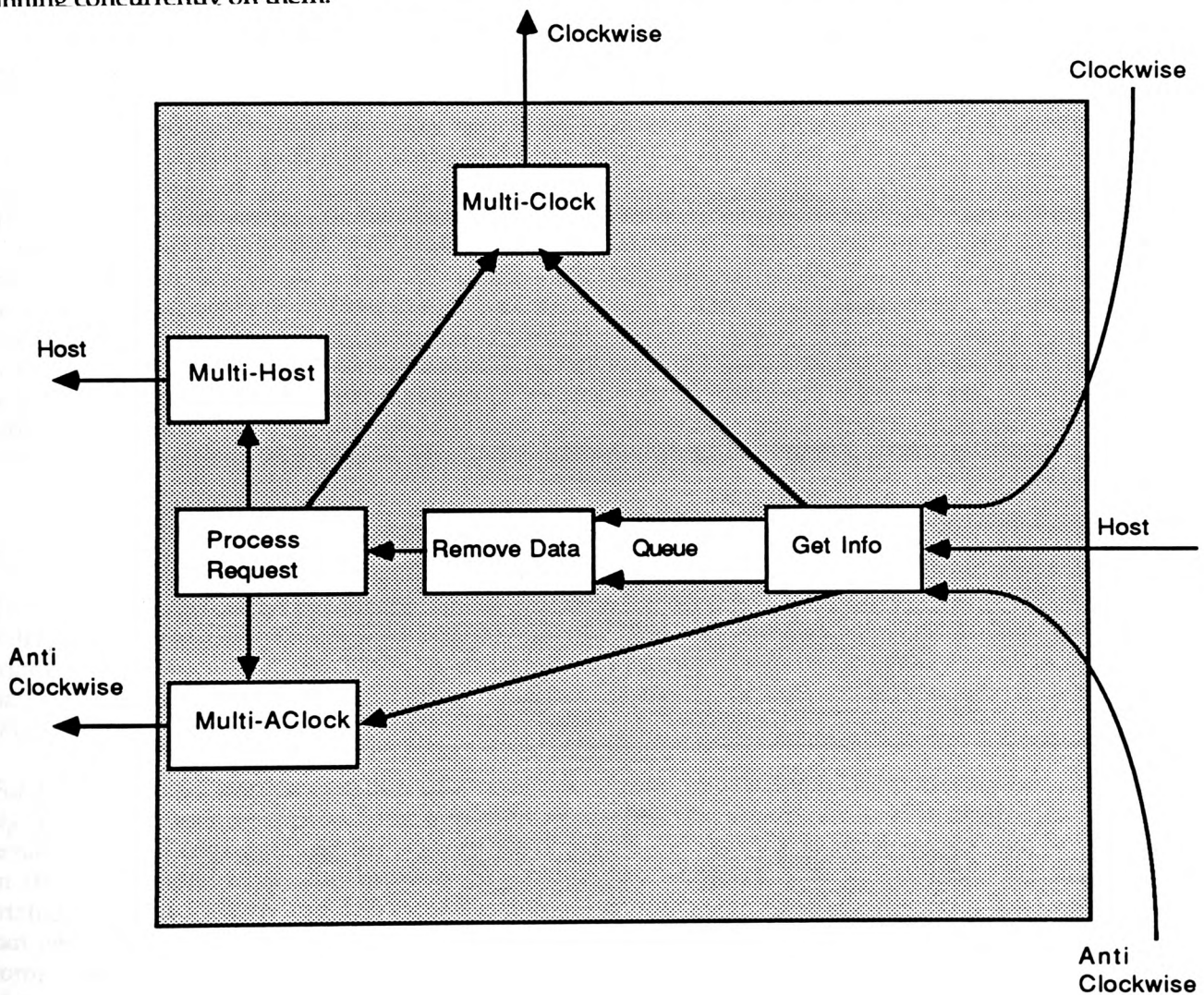


Figure 3 - Diagrammatical View Of The Processes Running Concurrently On The First Processor.

4. IMPLEMENTATION DETAILS

The initial process of most triangulation algorithms is to calculate the convex hull of the data. For a parallel implementation, this must be accomplished for the complete data set, rather than the

convex hull of each processor subset. In this prototype model, the vertices of the triangulation are derived from a regular grid DTM. As such, the boundary of the complete data region forms a rectangle, so the convex hull does not pose a problem. In addition, the boundary points are implicitly identified from their x or y coordinates. For a model such as this, the boundary points are not treated any different from interior points. At the triangulation stage, however, the boundary points enable the search for Thiessen neighbours to stop at the boundary with the 'outside world'.

The spatial search for data points is organised in a grid overlay or 'box-sort' auxiliary data structure (McCullagh & Ross, 1980; Kidner & Jones, 1991). This enables an efficient search for Thiessen neighbours to be limited to the most likely candidates. A grid overlay is used to spatially address all the points within each grid cell, provided that the data have been sorted by x and y coordinates. However, if the vertices are derived from a regular grid, the data can be written to a file in this order, without sorting. A further consideration in this parallel implementation is the relationship between the grid structure and the data subset for each processor. The main advantage of the grid is its simple and very efficient, direct access to points within a grid cell. Therefore, the partitioning of the vertices between processors should correspond with the nature of the spatial addressing of points. Since the data are organised within a grid, it follows that the assignment of processors should correspond to a number of grid cells, organised either in long, rectangular strips or in square blocks of cells, such as a quadtree.

For the prototype implementation, the rectangular strips were chosen, due to their greater flexibility. Strips of grid cells, corresponding to a row or column of the grid overlay provides a simple allocation of the data to processors, whilst maintaining an efficient spatial search for triangulation. The vertices in each processor strip can be accessed sequentially for the calculation of Thiessen neighbours, without searching at each stage for the next point within the subset. For this implementation, blocks of strips are assigned to each processor, since every processor is likely to access a number of strips. This also allows distinct data sets to be assigned to each processor, such that the complete data set need not be copied to each transputer. This would have been necessary with narrow strips, since a processor would not know in advance what strip it would be assigned and neighbouring strips would also have to be copied, since edges will connect points in neighbouring strips. The number of strips assigned to each processor is calculated such that the number of points is distributed as equally as possible, within the confines of the overlaying grid.

5. RESULTS AND CONCLUSIONS

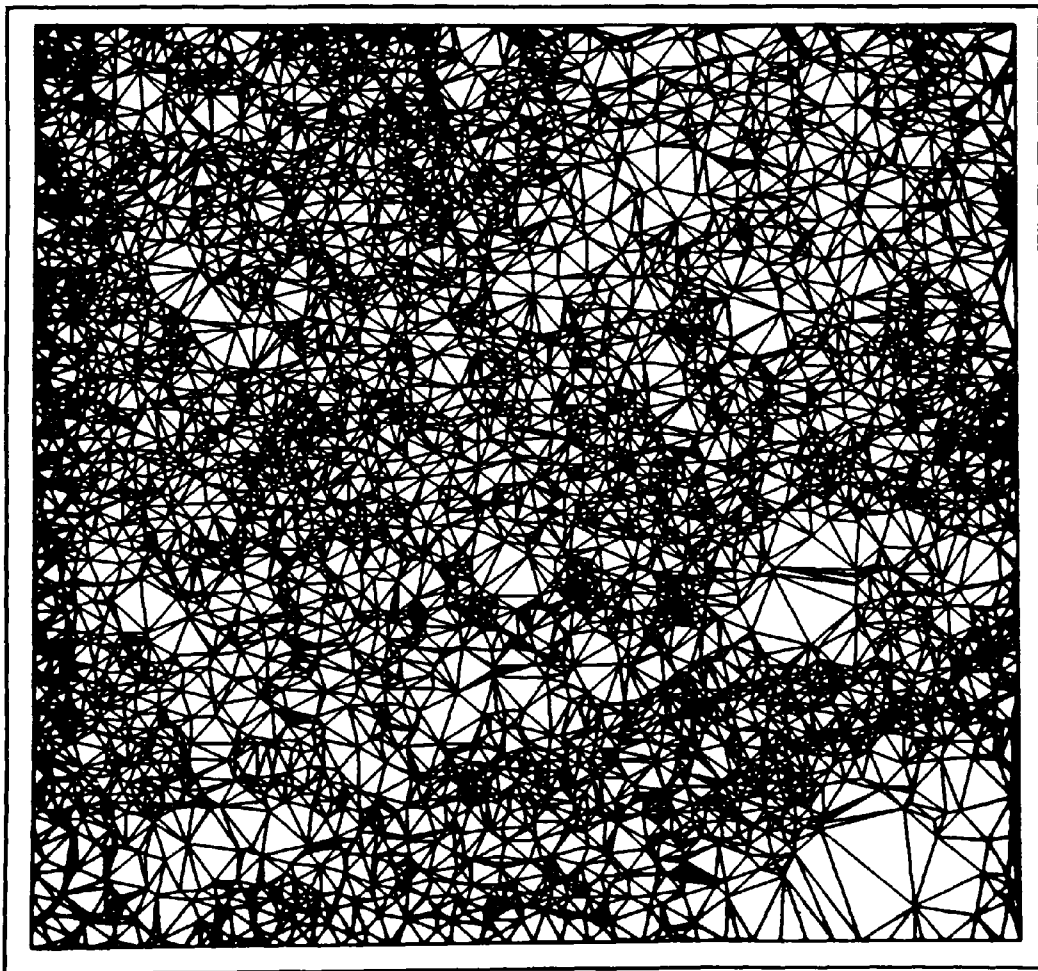
The test data set for the prototype model represents a 20 x 20 kilometre region of the South Wales valleys (National Grid Reference ST08/09/18/19). The vertices were initially derived by a dynamic triangulation, such that the TIN was constrained to a maximum error tolerance of 15 metres. The network consists of 8278 vertices (309 of which are on the boundary), 16,245 triangles and 24,522 edges (Figure 4).

For the prototype implementation of a network of four transputers, initial testing has shown that a significant reduction (approaching a quarter of the time for one processor) in processor time can be achieved when triangulating for a given set of points. Thus for this implementation, time decreases in direct proportion to the number of processors. Figures 5 & 6 illustrate the stages of the triangulation at 1/12 th and 1/6 th of the time taken for one processor. In effect, the first (top) strip can be considered to be the equivalent stage for just one processor in the same time allocation. A comparison of the TINs with the original in Figure 4, shows that the triangulation at each stage is identical. This proves that the Delaunay triangulation is unique, irrespective of initial starting point.

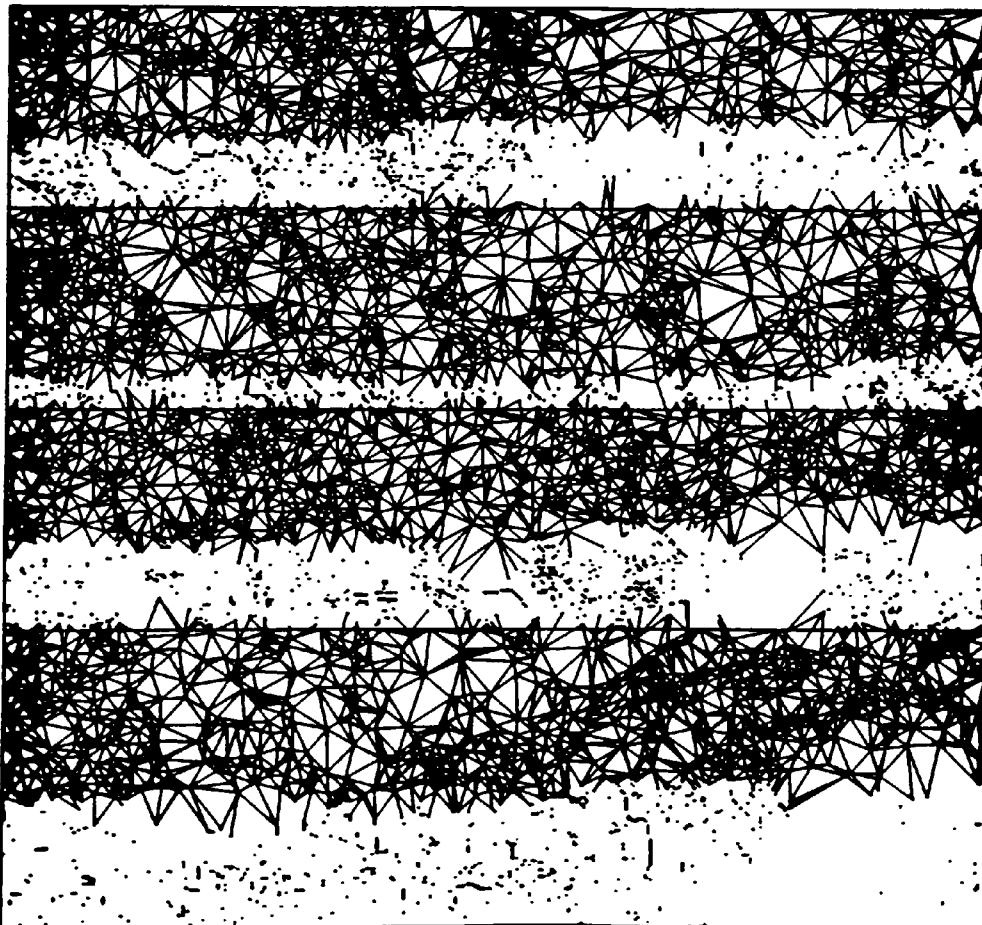
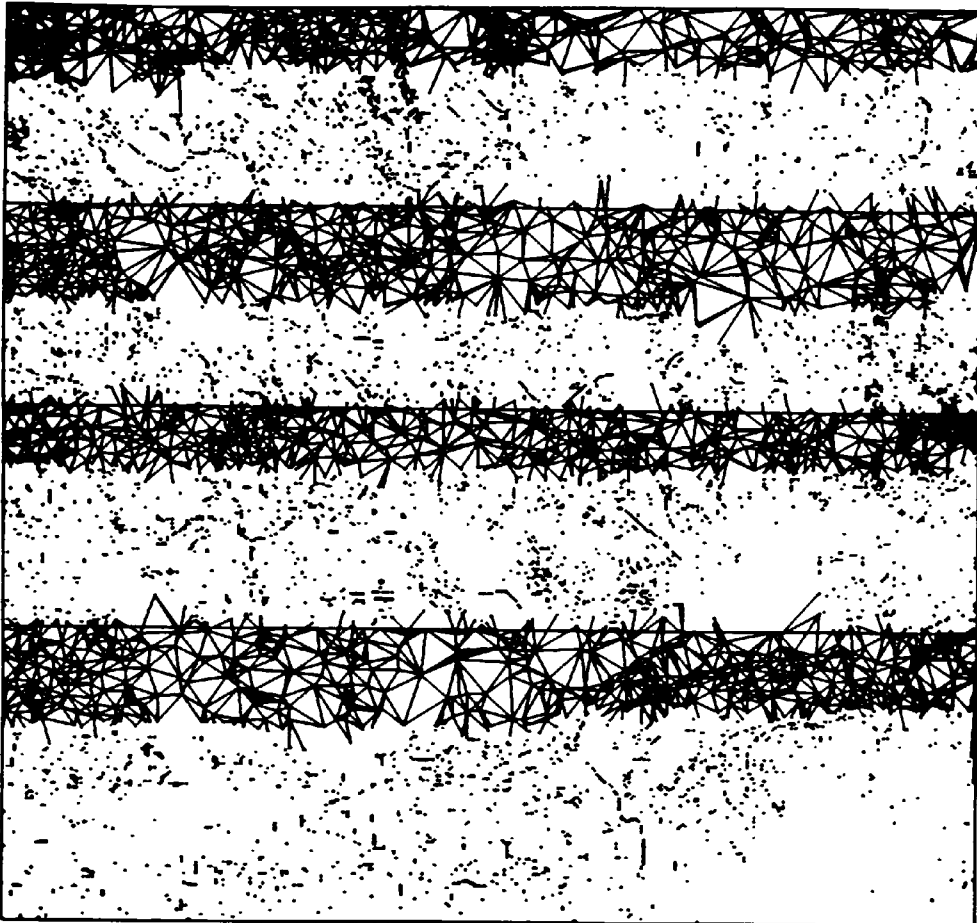
The bidirectional loop of four transputers can easily be extended to facilitate an even faster turn around rate (see work by Ware et al). The prototype implementation was therefore extended to a network of eight (2 PCBs of four transputers) and 16 (4 PCBs) transputers. The resulting TINs at 1/12 th and 1/24 th of the time taken for one processor are illustrated in Figures 7 & 8. The former of these figures is therefore directly comparable with Figure 5 after the same elapsed CPU time.

It should be noted however that the law of diminishing returns will come into force as the size of the network is increased. This is due to the increased communication overheads incurred when passing information around the network. Another factor which will affect the performance of the system is the partitioning of the data. Despite strips of approximately equal numbers of vertices being assigned to each processor, the time taken for each processor to calculate the Thiessen neighbours will be different. This variation, whilst small as a proportion of total time, has a tendency to be more significant as the number of processors increases. Since the effective total time is equivalent to the time of the last processor to finish, any significant variance could cause a noticeable decrease in performance, or 'processor redundancy'.

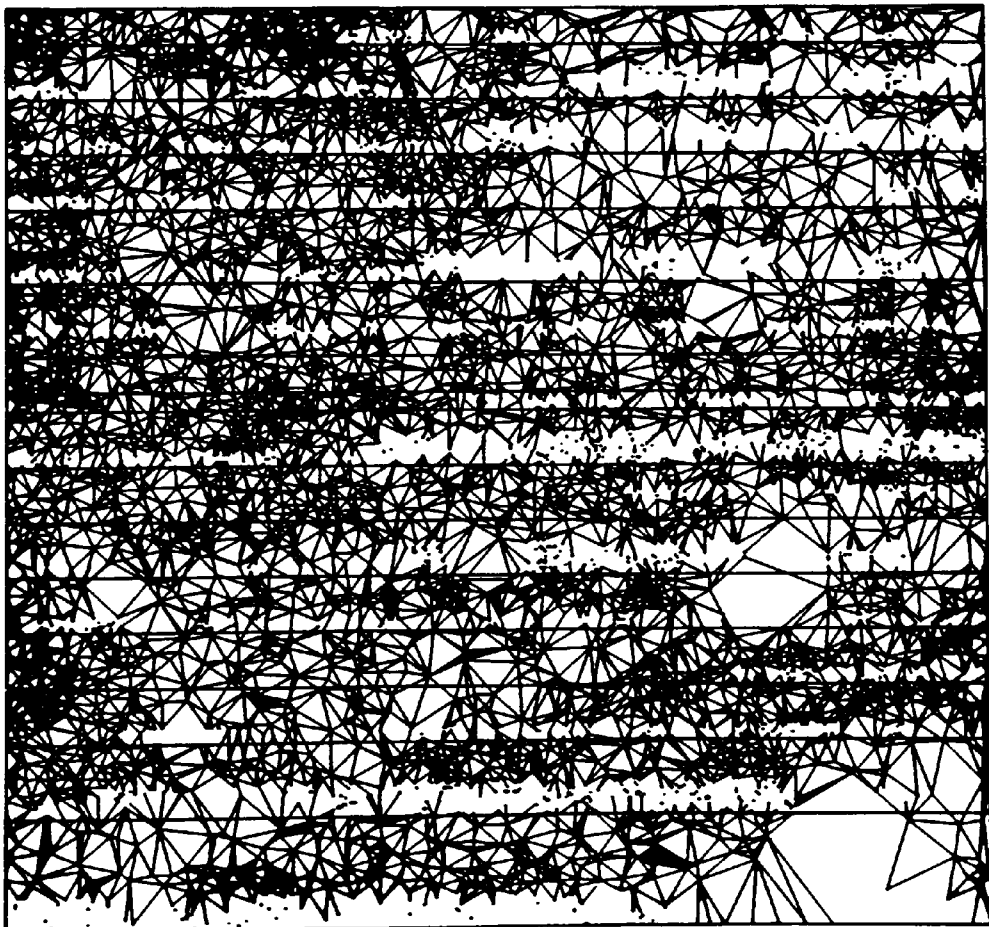
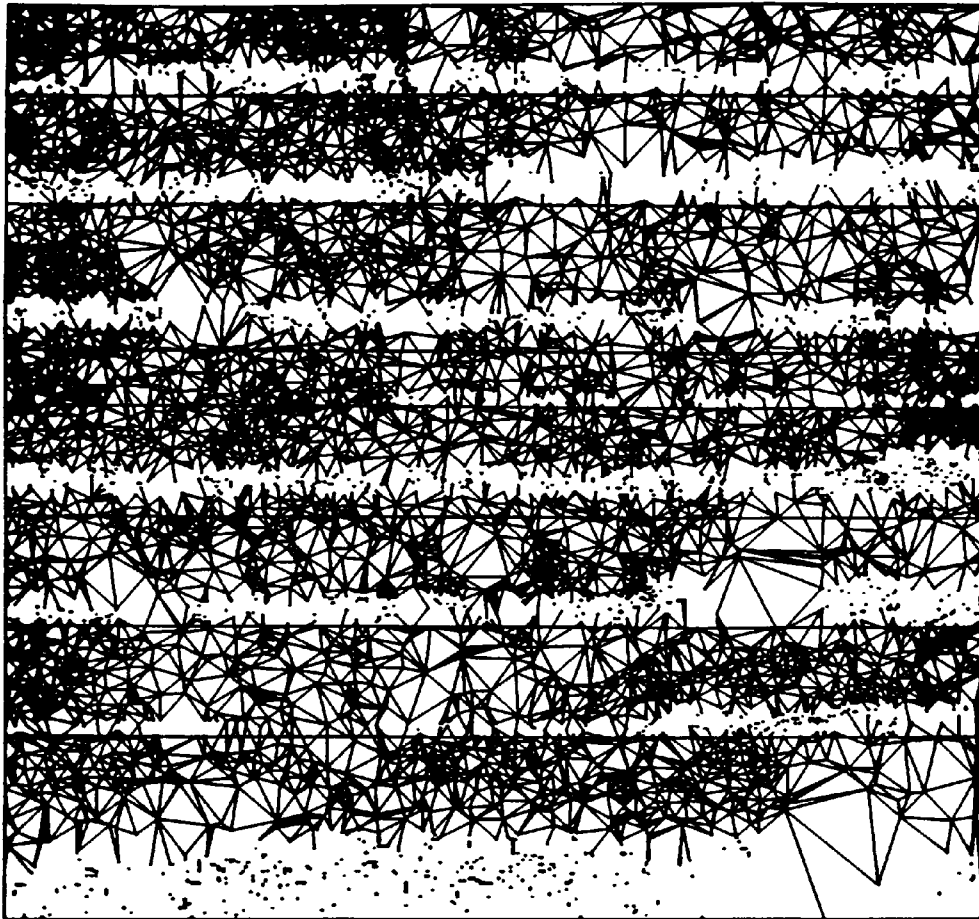
Any variation in time for the processors is due to the calculation of each vertex's Thiessen neighbours and more specifically the search time required to find these edges. An equal number of points per processor however, will not necessarily produce an equal number of edges. This will be variable, depending upon the spatial distribution of the data. Thus search time is not directly related to the number of vertices, but rather the number of edges. In the examples of Figures 5-7, for four and eight processors, the variation in number of calculated Thiessen neighbours (or edges) is over 1000, for a very small variation in vertices. This can be seen in Figure 6, where the second processor appears closest to finishing. Hence, an ideal partitioning of the data should be related to the number of edges per processor. However, this cannot be determined before triangulation has been completed, so an alternative criterion for data partitioning is required. This could possibly be accomplished by examining the density of point sampling within each possible processor strip. For the triangulation algorithm described in this paper, it is clear that the search time for each possible Thiessen neighbour is related to the number of points within each search circle. Hence, the searching algorithm should be more computationally efficient for sparser data.



*Figure 4 - Triangulated Irregular Network for Test Data Set ST08
(8278 Vertices, 309 Boundary Points, 16245 Triangles and 24522 Edges)*



Figures 5 and 6 - ST08 TIN Using Four Processors at $1/12$ th and $1/6$ th of CPU Time of One Processor



Figures 7 and 8 - ST08 TIN Using 8 and 16 Processors at $1/12$ th and $1/24$ th of CPU Time of One Processor

6. REFERENCES

- De Floriani, L. (1987) "Surface Representations Based on Triangular Grids", The Visual Computer, Vol.3, No.1, Feb., pp.27-50.
- El Gindy, H. (1986) "An Optimal Speed-Up Parallel Algorithm for Triangulating Simplicial Point Sets in Space", Int. Journal of Parallel Programming, Vol.15, No.5, Oct., pp.389-398.
- El Gindy, H. (1990) "Optimal Parallel Algorithms for Updating Planar Triangulations", Proc. of the 4th Int. Symposium on Spatial Data Handling, Zurich, Vol.1, July 23-27, pp.200-208.
- Gold, C.M. (1979) "Triangulation-Based Terrain Modelling - Where Are We Now?", Proc. of Auto Carto IV, Vol.2, Nov.4-8, pp.104-111.
- Heller, M. (1990) "Triangulation Algorithms for Adaptive Terrain Modeling", Proc. of the 4th Int. Symposium on Spatial Data Handling, Zurich, Vol.1, July 23-27, pp.163-174.
- Kidner, D.B., Jones, C.B. (1991) "Implicit Triangulations For Large Terrain Databases", Proc. of the Second European Conference on Geographical Information Systems.
- Lee, D.T., Schachter, B.J. (1980) "Two Algorithms For Constructing a Delaunay Triangulation", Int. Journal of Computer and Information Sciences, Vol.9, No.3, pp.219-242.
- McCullagh, M.J., Ross, C.G. (1980) "Delaunay Triangulation of a Random Data Set for Isarithmic Mapping", The Cartographic Journal, Vol.17, No.2, Dec., pp.93-99.
- McCullagh, M.J. (1987) "Digital Terrain Modelling and Visualisation", Proc. of a Short Course in 'Terrain Modelling in Surveying and Civil Engineering', Univ. of Surrey, Apr.7-9, and Univ. of Glasgow, Sept.1-3, 27 pages.
- Merks, E. (1986) "An Optimal Parallel Algorithm For Triangulating A Set of Points in the Plane", Int. Journal of Parallel Programming, Vol.15, No.5, Oct., pp.399-411.
- Peucker, T.K., Fowler, R.J., Little, J.J., Mark, D.M. (1978) "The Triangulated Irregular Network", Proc. of the Digital Terrain Models (DTM) Symposium, May 9-11, pp.516-540.
- Ware J.A., Roberts G., Davies R.A., Miles R., Williams J.H. (1990) "A Modular Sensing System For Robotic Control", The Second Int. Conference On Applications Of Transputers, Southampton University, July, pp 78-85.

ACKNOWLEDGEMENTS

This work has been partially funded by the Royal Signals and Radar Establishment (R.S.R.E), Ministry of Defence, Procurement Executive, Malvern, Worcestershire, England.

COMPRESSION OF DIGITAL ELEVATION MODELS BY HUFFMAN CODING

D. B. KIDNER

Department of Computer Studies, The Polytechnic of Wales, Pontypridd,
Mid Glamorgan, Wales, U.K. CF37 1DL

D. H. SMITH

Department of Mathematics and Computing, The Polytechnic of Wales, Pontypridd,
Mid Glamorgan, Wales, U.K. CF37 1DL

ABSTRACT

Regular rectangular grid digital elevation models (DEMs) are the most popular means of representing terrain data, due primarily to their simplicity, implicit coordinates, application efficiency and widespread availability of data in this format. However, in many instances, they are not storage-efficient, since a regular sampling of points is not adaptive to the variability of terrain, resulting in data redundancy in flatter regions. Data redundancy also occurs when elevations are represented by storage units of fixed size, such as two-byte integers. An alternative representation is described which attempts to remove this redundancy, thus minimising the storage requirements of DEMs. This saving can be as much as 80% to 90% (depending on terrain variability), with no loss of accuracy. This is accomplished by predicting the grid elevations and storing the errors or differences using Huffman codes. There is an overhead of additional time in reconstructing the DEM from its compressed form. The effect of incorporating error tolerances into these grid DEMs has also been investigated. An algorithm with FORTRAN 77 program is presented which calculates the Huffman codes for the elevations of a given DEM.

Key Words: Digital elevation model, Digital terrain model, Data compression, Statistical data encoding, Huffman coding.

INTRODUCTION

Digital terrain models may be defined as the numerical (or digital) and mathematical representation of a terrain by making use of adequate elevation measurements, which are compatible in number and distribution with that terrain, so that the elevation of any point of known coordinates can be automatically interpolated with required specified accuracy for any given application (Ayeni, 1978). The term digital elevation model (DEM) is also commonly used, but because the term 'terrain' often implies attributes of landscape other than the altitude of the land-surface, including derived data about the terrain such as slope, aspect, visibility, etc., the term DEM is preferred for models containing only elevation data (Petrie & Kennie, 1987).

Most DEMs are generated from photogrammetric measurements of the terrain, by correlating aerial photographs using analytic stereo-plotting machines. These stereo models are then scanned very accurately to produce rows of elevation coordinates, which can be output in the form of regular grid cells, usually sampled at 30 or 50 metre intervals. Other data acquisition techniques, such as digitising existing contour maps, or terrestrial surveys, will usually result in the arbitrary sampled data being interpolated onto a regular grid (Petrie, 1987). These square or rectangular grids, stored in a two-dimensional matrix, are the most widely used of all DEM data structures, since their simplicity leads to convenient programming; the topology is implicit, with no need to store the x,y coordinates; and their spatial symmetry allows minimal search time for applications. The obvious shortcoming of the method is its inability to adapt to the variability of the terrain, resulting in data redundancy, especially in areas of flat, uniform terrain. The problem is exaggerated by the fact that national mapping agencies produce DEMs at a fixed grid resolution to obtain sufficient accuracy and compatibility between models.

Grid DEMs are conventionally stored as two-dimensional matrices, either as 8 bits, 16 bits or 32 bits for each sampled value. However, the choice of storage unit size is often arbitrary, without regard to how much data is sufficient to represent what is known (Dutton, 1983). Since the range of grid elevations in a DEM will usually be greater than 256 metres, eight bits of storage will be insufficient, unless the data is quantised into elevation classes (with a resulting loss of accuracy). Therefore, 16 bits of storage for each value is typical of most DEMs. However, the 65,536 possible classes are rarely utilised, since the accuracy to which the terrain can be sampled cannot be guaranteed by the data acquisition method. For example, a DEM with an elevation range of 1000 metres could be represented to within a tolerance of 2 centimetres, which is beyond the scope of accuracy for existing, data collection techniques. This redundancy is increased further by the tendency to use elevations rounded to the nearest integer value. Elevations for the Ordnance Survey (O.S.) grid DEMs (sampled every 50 metres in x and y) are represented to the nearest metre, whilst for the United States Geological Survey (U.S.G.S.) 7.5-minute DEMs (sampled every 30 metres in x and y), elevations are represented to the nearest foot or metre, depending on the relief (USGS, 1990).

These two aspects of data redundancy (due to terrain variability and fixed-size matrix storage), can prove costly for users who require access to large areas of topography (for example, mapping or defence organisations). Whilst storing an individual DEM in its full two-byte matrix format in computer memory may not be a problem to many users today, the cost of storing tens, hundreds, or thousands of DEMs on secondary storage may well be exorbitant. Similarly, the small microcomputer user will also experience high storage costs in representing DEMs.

These problems have led to the increasing popularity of alternative terrain models, such as the triangulated irregular network (TIN). The TIN is not only adaptive to terrain variability, but storage can be substantially reduced when compared to the regular grid, with the adoption of an efficient data structure (Kidner & Jones, 1991). However, many applications can be performed more efficiently with the regular grid DEM.

The method proposed in this paper, aims to minimise the storage overheads of the regular grid by using a local technique to predict elevations in the DEM; the errors for which are compressed using Huffman coding. It is these coded errors or corrections that are then stored, rather than the elevations themselves. This removes the redundancy in the data, since the variable-length codes are assigned according to the variability of the terrain.

STATISTICAL DATA ENCODING

Despite the adaptability of some local techniques, mathematical modelling is ill-suited for representing topographic data, due to its tendency of 'smoothing out' terrain characteristics, whilst at the same time, generating false anomalies (Williams, 1986). An alternative approach is to identify the redundancy in the data storage, and to improve this efficiency by the application of a practical data compression routine. These techniques are used mostly in the field of data communications, but the same theory holds for increasing the capacity of mass storage devices.

As already mentioned, the data redundancy of grid DEMs exists because elevations are stored as fixed-length values (eg. 16 bits), and no distinction is made between elevations in the terrain, whether variable or not. The ideal approach is to assign variable-length codes to heights, according to some statistical criterion relating elevations. This statistical encoding would take advantage of the probabilities of occurrence of symbols (ie. the relationship between elevations), so that short codes can be used to represent frequently occurring symbols, while longer codes are used to represent less-frequently encountered symbols (Held, 1987). Thus the total number of bits constituting the data can be minimised. Such a code is known as a frequency-dependent code; one example of which is the Huffman coding technique (Huffman, 1952).

Before discussing this method, some basic information theory will provide an understanding of how redundancy can be statistically reduced. For any variable-length coding scheme, the number of different characters or symbols is n . If $p(i)$ is the probability of the i th symbol, then

$$\sum_{i=1}^n p(i) = 1.00. \quad (1)$$

The length of a message, $l(i)$ is the number of coding digits (bits) assigned to it. Therefore, the average message length is:

$$l_{ave} = \sum_{i=1}^n p(i) l(i) \quad (2)$$

The average information per single symbol is:

$$H_{ave} = - \sum_{i=1}^n p(i) \log_2 p(i) \text{ bits/symbol.} \quad (3)$$

This equation represents the mathematical definition of entropy, a term used in information theory to denote the average number of bits required to represent each symbol of a source alphabet. (For the derivation of this, see Held, 1987). The efficiency of a code can therefore be calculated as:

$$\text{EFFICIENCY} = (H_{ave} / l_{ave}) \times 100\%. \quad (4)$$

HUFFMAN CODING

Huffman (1952) developed a procedure for encoding a statistically independent source in such a way as to yield the minimum average code length, or most efficient code. This minimum-redundancy code or optimum code, has some basic restrictions imposed on its construction:

- (a) No two symbols consist of identical arrangements of coding digits.
- (b) The symbol codes are constructed in such that no additional indication is necessary to specify where a symbol code begins and ends, once the starting point of the sequence is known.
- (c) The length of a given code can never be less than the length of a more probable code. Thus,

$$p(1) \geq p(2) \geq p(3) \geq \dots \geq p(n-1) \geq p(n) \quad \text{and} \quad (5a)$$

$$l(1) \leq l(2) \leq l(3) \leq \dots \leq l(n-1) = l(n) \quad \text{for an optimum code.} \quad (5b)$$

The code that will be considered here is a binary (two-state) symbol code. Thus to use Huffman's algorithm, a binary coding tree is constructed as follows:

- (i) Arrange the source probabilities of the symbols (nodes) in descending order.
- (ii) Commencing with the symbols with the two lowest probabilities p_a and p_b , construct a new node of which these two probabilities are branches, the new node being labelled with the arithmetic sum of these two probabilities.
- (iii) Repeat the process using the new node instead of the original two, until only one node is left, with a label probability of 1.00.
- (iv) Label each upper branch with a '0' and the lower member of each pair with a '1', or vice versa.
- (v) The code for each of the original symbols is then found by proceeding from the root of the tree to the required leaf, noting the branch label of each node traversed.

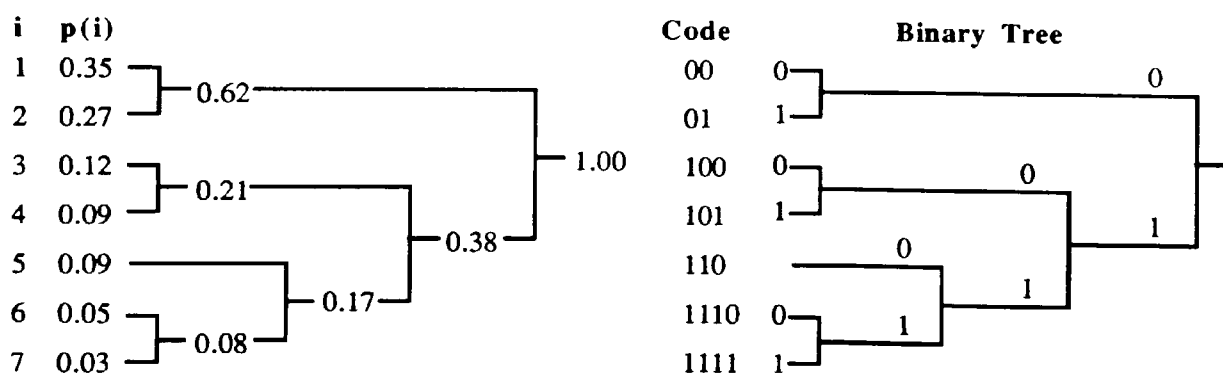


Figure 1. Formation of Huffman codes of a probability distribution.

For example, consider the following code of seven symbols, with the probabilities of occurrence 0.35, 0.27, 0.12, 0.09, 0.09, 0.05 and 0.03. The Huffman codes for these probabilities are calculated in Figure 1. A more detailed analysis of this coding scheme is presented in Table 1. It can be seen that

the average code length is 2.46 bits/symbol (from equation 2), whereas the entropy is 2.4005 bits/symbol (equation 3). Thus, from equation (4), the efficiency of this Huffman code is 97.5813% and equation (5) is satisfied for an optimum code. Using a fixed-length code, the seven symbols would require a minimum of three bits of storage for the eight possible states, the efficiency of which is only 80.0167%. However, as the size of the data set increases, the efficiency of a fixed-length coding scheme will decrease substantially, whilst the efficiency of Huffman coding may improve significantly for a suitable data probability distribution.

i	p(i)	Code	l(i)	p(i)l(i)	-p(i)log ₂ p(i)
1	0.35	00	2	0.70	0.5301
2	0.27	01	2	0.54	0.5100
3	0.12	100	3	0.36	0.3671
4	0.09	101	3	0.27	0.3127
5	0.09	110	3	0.27	0.3127
6	0.05	1110	4	0.20	0.2161
7	0.03	1111	4	0.12	0.1518
Sum =				2.46	2.4005
Bits / Symbol					

Table 1. Huffman Codes, Code Length and Entropy for the Data in Figure 1.

To decode a Huffman-encoded data set, each bit is read in turn, checking at each stage to see whether a valid codeword has been detected. A list of codes and their corresponding decoded symbol values are therefore required to check whether the encoded message is a matched codeword. If it is, then its corresponding value can be returned as the decoded symbol. The next bit to be read will be the start bit of the next codeword. Otherwise, if there is no matching codeword, the encoded message is extended one bit at a time, until a match is found. To facilitate this procedure, the search time can be reduced significantly by only checking those codewords in the list which have the same number of bits as the encoded message. This can be achieved with a look-up table indicating the first position in the list of codewords of values with the same bit-length. Hence, only a few codewords, if any, will have to be checked at each stage. An example of a look-up table for a list of codewords is shown in Figure 2, for the example illustrated in Figure 1.

Number of Bits	Code Frequency	Pointer	Code	Symbol
1	0	0	00	1
2	2	1	01	2
3	3	3	100	3
4	2	6	101	4
			110	5
			1110	6
			1111	7

Figure 2. Decoding Table and Look-Up Table for the Huffman Codes in Figure 1.

To illustrate how this is used in the decoding process, consider an encoded message corresponding to the Huffman example in Figure 1:

11000101010011101001001111110011110011110101101

The first bit is read (1), but from the look-up table in Figure 2, it is noticeable that there are no codewords with a bit length of one. Hence, no search is necessary. The next bit is read (1), such that the encoded symbol is now 11. From the look-up table, it can be seen that there are two codes of length two bits, starting at position one in the codeword table. However, since these do not match, the encoded symbol is extended by another bit (0) to become 110. There are three codes of length three bits in positions three to five in the codeword table. A match is found at position five for the symbol '5'. Decoding then continues for the next symbol. A summary of this process for the first three codewords is:

(i)	Message = 1	Bits = 1	Frequency = 0	No Search	No Match
	Message = 11	Bits = 2	Frequency = 2	Search 1-2	No Match
	Message = 110	Bits = 3	Frequency = 3	Search 3-5	Match Symbol = '5'
(ii)	Message = 0	Bits = 0	Frequency = 0	No Search	No Match
	Message = 00	Bits = 2	Frequency = 2	Search 1-2	Match Symbol = '1'
(iii)	Message = 1	Bits = 1	Frequency = 0	No Search	No Match
	Message = 10	Bits = 2	Frequency = 2	Search 1-2	No Match
	Message = 101	Bits = 3	Frequency = 3	Search 3-5	Match Symbol = '4'

This procedure is repeated until the complete message is decoded as:

110-00-101-01-00-1110-100-100-1111-110-01-1110-01-1110-101-101

or symbols 5-1-4-2-1-6-3-3-7-5-2-6-2-6-4-4.

HUFFMAN-ENCODED DEMs

A grid or two-dimensional matrix of elevations is not the ideal choice for Huffman encoding, since the frequency or probability distribution of heights will be fairly uniform, resulting in codewords being of similar bit-length. The number of codes will also be equal to the number of different elevations, which could be a few hundred. Huffman coding is at its most efficient when applied to a distribution of values that approximates a normal distribution. The problem can be overcome by considering the differences in elevation from predicted height values, instead of the elevations themselves. In this way, a distribution of elevation differences can be formed, which actually takes into account the variability of the terrain.

The most common approach to forming an elevation difference grid is to store the first height of each row (or column), and then calculate the difference between this and the next elevation. This continues to the end of the row, and repeated for all the rows in the grid. An example of this is shown below in Figures 3 and 4 for a 16x16 sub-grid of the Ordnance Survey (O.S.) grid reference ST08 DEM, sampled at 50 metre intervals in x and y. The enclosed 15x15 grid region is the area for which the differences are calculated.

Row 15	338	333	332	327	337	343	347	352	358	369	375	381	393	405	413	420
14	352	349	345	339	350	351	361	359	360	365	373	378	386	397	406	413
13	368	364	359	354	360	364	369	367	364	369	372	378	385	393	401	408
12	382	375	368	363	369	372	375	377	373	371	375	379	384	390	397	402
11	394	386	378	373	375	378	381	384	385	380	379	382	385	390	394	399
10	407	396	388	381	381	384	386	389	391	390	384	390	390	390	393	395
9	418	409	399	392	390	390	391	392	392	391	388	390	390	391	391	392
8	426	420	409	403	399	398	398	396	395	393	390	390	390	391	390	389
7	435	428	422	414	410	405	402	399	398	395	391	389	387	385	386	385
6	448	443	433	425	419	410	407	402	401	398	393	388	383	380	378	379
5	454	453	447	441	432	426	420	409	405	401	395	388	382	377	370	372
4	457	457	454	450	444	438	430	419	408	404	398	390	384	378	370	366
3	459	460	459	458	451	448	442	431	417	408	402	393	387	381	372	366
2	459	461	463	462	459	456	452	442	427	414	405	396	388	382	374	367
1	458	462	465	468	465	461	457	449	438	421	409	397	389	382	375	368
	454	461	467	470	470	464	461	454	445	430	413	400	389	383	375	369

Figure 3. 16x16 Subset of Original Grid DEM (O.S. ST08).

Row 15	338	-5	-1	-5	10	6	4	5	6	11	6	6	12	12	8	7
14	352	-3	-4	-6	11	1	10	-2	1	5	8	5	8	11	9	7
13	368	-4	-5	-5	6	4	5	-2	-3	5	3	6	7	8	8	7
12	382	-7	-7	-5	6	3	3	2	-4	-2	4	4	5	6	7	5
11	394	-8	-8	-5	2	3	3	3	1	-5	-1	3	3	5	4	5
10	407	-11	-8	-7	0	3	2	3	2	-1	-6	6	0	0	3	2
9	418	-9	-10	-7	-2	0	1	1	0	-1	-3	2	0	1	0	1
8	426	-6	-11	-6	-4	-1	0	-2	-1	-2	-3	0	0	1	-1	-1
7	435	-7	-6	-8	-4	-5	-3	-3	-1	-3	-4	-2	-2	-2	1	-1
6	448	-5	-10	-8	-6	-9	-3	-5	-1	-3	-5	-5	-5	-3	-2	1
5	454	-1	-6	-6	-9	-6	-6	-11	-4	-4	-6	-7	-6	-5	-7	2
4	457	0	-3	-4	-6	-6	-8	-11	-11	-4	-6	-8	-6	-6	-8	-4
3	459	1	-1	-1	-7	-3	-6	-11	-14	-9	-6	-9	-6	-6	-9	-6
2	459	2	2	-1	-3	-3	-4	-10	-15	-13	-9	-9	-8	-6	-8	-7
1	458	4	3	3	-3	-4	-4	-8	-11	-17	-12	-12	-8	-7	-7	-7
	454	461	467	470	470	464	461	454	445	430	413	400	389	383	375	369

Figure 4. 16x16 Subset and 15x15 Row Elevation Differences (Corrections).

The elevations of the original grid can then be recalculated from these differences or error corrections. For example, the first row can be reconstructed from its base (first) elevation of 458 metres by simply adding the differences at each stage (ie. $458+4=462$, $462+3=465$, $465+3=468$, $468-3=465$, etc.). It can be seen from the grid in Figure 4, that the range of values used to represent (or reconstruct) the terrain model is smaller than the range of the original elevations.

A number of different methods were examined to try and improve upon this performance, ie. to get a narrower range of values. These methods aim to predict the neighbouring elevations, instead of using the differences between neighbours. They included slope (or gradient) analysis and polynomial modelling techniques. However, the method which gave the best overall results was a prediction or extrapolation technique for a node based on the three neighbouring points forming a grid cell. This simple method assumes that the linearly interpolated mid-point between a pair of diagonal grid nodes will be equivalent in elevation to the linearly interpolated mid-point of the other intersecting diagonal forming the grid square. This is illustrated in Figure 5.

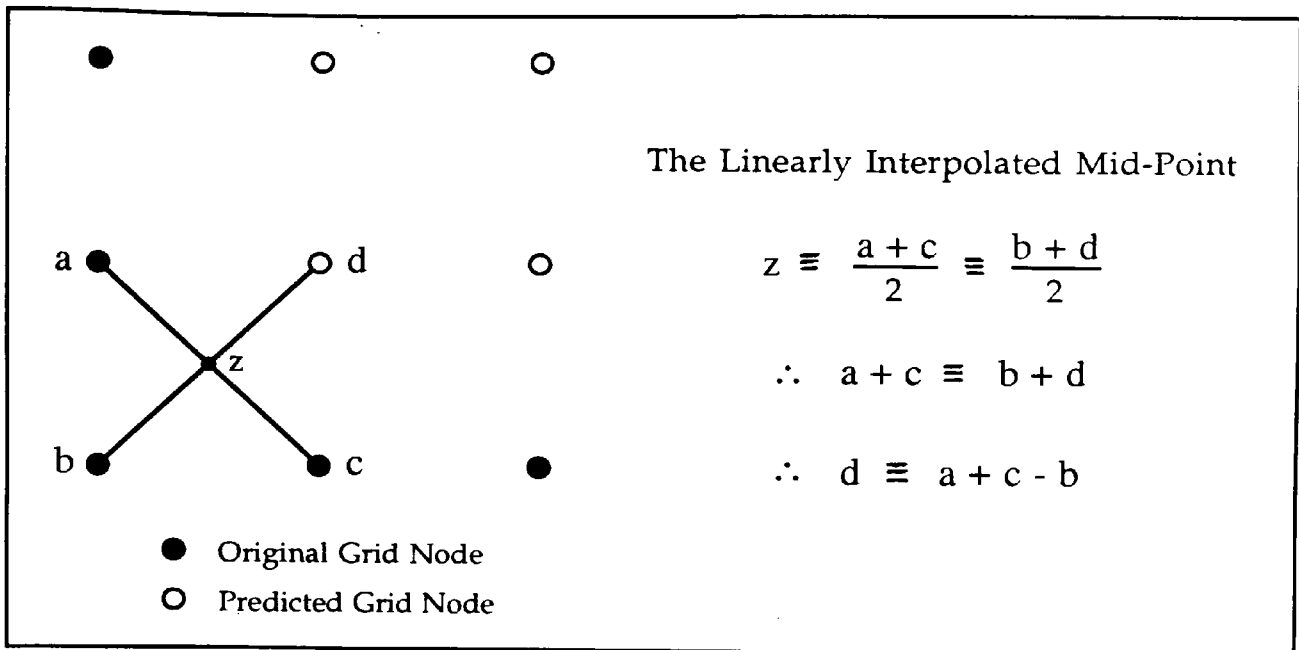


Figure 5. Algorithm to Predict Regular Grid Elevation (d) Using an Extrapolation Procedure Based Upon the Intersecting Diagonals of a Grid Cell.

The difference in elevation between this height (d) and the actual elevation is used as the correction for Huffman coding. This process is repeated along the row, for all rows in the grid. This method assumes that the first row and column of each grid DEM are known, rather than the first row or column of the difference method. When applied to the grid DEM in the above example, the 'predicted' differences are shown below in Figure 6.

Row 15	338	- 2	3	1	- 1	5	- 6	7	5	6	- 2	1	4	1	- 1	0
14	352	1	1	- 1	5	- 3	5	0	4	0	5	- 1	1	3	1	0
13	368	3	2	0	0	1	2	- 4	1	7	- 1	2	2	2	1	2
12	382	1	1	0	4	0	0	- 1	- 5	3	5	1	2	1	3	0
11	394	3	0	2	2	0	1	0	- 1	- 4	5	- 3	3	5	1	3
10	407	- 2	2	0	2	3	1	2	2	0	- 3	4	0	- 1	3	1
9	418	- 3	1	- 1	2	1	1	3	1	1	0	2	0	0	1	2
8	426	1	- 5	2	0	4	3	1	0	1	1	2	2	3	- 2	0
7	435	- 2	4	0	2	4	0	2	0	0	1	3	3	1	3	- 2
6	448	- 4	- 4	- 2	3	- 3	3	6	3	1	1	2	1	2	5	- 1
5	454	- 1	- 3	- 2	- 3	0	2	0	7	0	0	1	0	1	1	6
4	457	- 1	- 2	- 3	1	- 3	- 2	0	3	5	0	1	0	0	1	2
3	459	- 1	- 3	0	- 4	0	- 2	- 1	1	4	3	0	2	0	- 1	1
2	459	- 2	- 1	- 4	0	1	0	- 2	- 4	4	3	3	0	1	- 1	0
1	458	- 3	- 3	0	- 3	2	- 1	- 1	- 2	- 2	5	1	3	- 1	1	- 1
	454	461	467	470	470	464	461	454	445	430	413	400	389	383	375	369

Figure 6. 16x16 Subset and 15x15 Elevation Differences Using Linear Prediction.

It can be seen that the range of elevation differences has been reduced significantly, with a resulting better probability (occurrence) distribution for Huffman encoding. A summary of these difference-encoded grids (Table 2 and Figure 7) for the data in Figure 3, highlights this fact. The graph of probability distributions for these methods (Figure 7), illustrates clearly that the linearly predicted elevations give the best distribution of corrections for Huffman encoding.

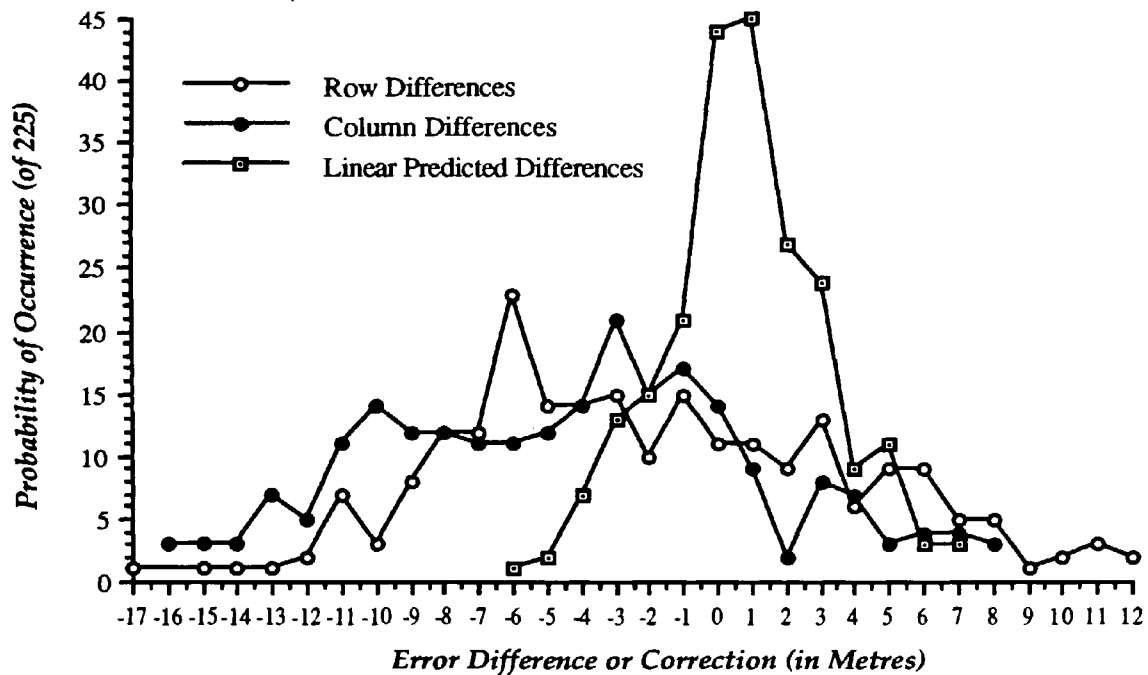


Figure 7. Frequency Distribution of Elevation Corrections for (a) Row (b) Column and (c) Linear Predicted Differences of 15x15 Grid DEM in Figure 1.

Elevation Distribution	Range (m)	Values (Codes)	Largest Occurrence (of 225) and Probability (of 1.0)
Original Elevations	327 to 470	110	15 - 0.06667
Column Differences	-16 to 8	25	21 - 0.09333
Row Differences	-17 to 12	29	23 - 0.10222
Linear Predicted Differences	-6 to 7	14	45 - 0.20000

Table 2. Range, Number of Codes and Largest Probability for the 15x15 Grid DEM.

GRID DEM RESULTS

The elevation differences or corrections of the linear predictor method were Huffman-encoded for three contrasting DEMs. The first two DEMs represent Ordnance Survey (O.S.) 20 x 20 km grids of South Wales, referenced as ST06 (Figure 8) and ST08 (Figure 9) in Appendix A, sampled at 50 metre intervals of x and y (ie. 401x401 values), and rounded to the nearest metre interval. The third data set is a United States Geological Survey (U.S.G.S) 7.5-minute quadrangle DEM, with elevations sampled to the nearest foot at 30 metre intervals of x and y. This 11.28 x 13.83 kilometre data set represents part of the Tiefort Mountain Range in California (Figure 10, Appendix A).

The first grid ST06 (Figure 8), highlights the typical data redundancy of DEMs supplied by mapping agencies in providing model compatibility based on national grid coordinates - approximately one third of the region is represented by sea level values (the Bristol Channel). The

terrain itself is relatively flat with some steep slopes. The second model, ST08 (Figure 9), is typical of much of the terrain in South Wales (ie. mountains cut by steep valleys). The third model (Figure 10), represents a gently sloping plain culminating in a steep mountain.

For each DEM, it is assumed that the first row and column are known. (These values could also be Huffman-encoded, with the codes representing the elevation differences between adjacent grid nodes. As such, only the first elevation needs to be represented in full, whilst the codes would be the same as for the linear prediction method). Such an approach could also be used to code the irregular length profiles of the U.S.G.S. DEMs, along the extremes of the quadrangle. However, to keep the Huffman-encoding algorithm simple, these issues will not be described in the computer implementation (Appendix B) or the overall results.

Hence, the method was applied to the 400x400 grids of ST06 and ST08, and the 376x461 grid of the Tiefert Mountains. For this latter model, the algorithm was applied to the original elevations (in feet) and to the quantised grid of elevations in metres (ie. division by 3.28084 and rounded to the nearest integer). This provides four DEMs, the first three will all be error-free (in relation to the original accuracy), whilst the latter will include some rounding errors, the greatest of which is ± 0.5 metres. The results of Huffman-encoding these DEMs are presented in Table 3. In each case, there is a small additional overhead in storing the codewords and look-up table.

Surface Statistic	ST06	ST08	Tiefert Mountains (i)	Tiefert Mountains (ii)
Elevation Range	0 to 135 m	19 to 470 m	1519 to 4995 ft	463 to 1522 m
Average Elevation	31.32 m	173.17 m	2646.52 ft	806.94 m
Standard Deviation	34.39 m	88.43 m	476.83 ft	145.34 m
Linear Predicted Diff. Range	-40 to 41 m	-39 to 49 m	-35 to 41 ft	-11 to 13 m
Number of Huffman Codes	52	59	74	25
Average Code Length	1.9681 bits	3.2375 bits	3.3187 bits	2.2536 bits
Entropy	1.8810 bits	3.1830 bits	3.2765 bits	2.2198 bits
Efficiency of Code	95.5772 %	98.3154 %	98.7301 %	98.4977 %
Storage Saving (v. 2-Bytes)	87.6995 %	79.7654 %	79.2581 %	85.9150 %

Table 3. Surface Statistic Results for Huffman-Encoded DEMs.

The corresponding storage savings of Huffman-encoding the column and row elevation differences are 83.41% & 84.35% for ST06; 70.60% & 71.03% for ST08; 67.43% & 64.95% for Tiefert Mountains (in feet); and 76.98% & 75.12% for Tiefert Mountains (in metres). Hence, the results in Table 3 show that further significant storage savings can be attained with the application of a simple prediction algorithm.

The time taken to encode and decode the grid DEMs is also related to the variability of the terrain. As such, ST06 requires approximately 3 seconds of C.P.U. time to be reconstructed, whilst the corresponding times for ST08 and the Tiefert Mountain DEM are 5 and 6 seconds respectively. These timings are approximate, and serve only as a rough guide. Hence, when representing a DEM, the user is given the choice of storing a conventional grid DEM with its high storage overheads, or the Huffman-encoded grid DEM with its relatively high retrieval overheads (ie. STORAGE EFFICIENCY v. COMPUTATIONAL EFFICIENCY OF DEM RETRIEVAL). However, DEM retrieval is only a one-off process requiring a few seconds of C.P.U. time. For most users, this is an acceptable overhead when compared to the storage savings that can be achieved.

CONCLUDING REMARKS

The use of an error-free, Huffman-encoding algorithm based on linear prediction, enables storage savings of between 79% and 88% for typical regular grid DEMs, when compared to the storage requirements of conventional two-byte elevations. Whilst Huffman-encoding is not new (Huffman, 1952; Lynch, 1985; Williams, 1986; Held, 1987), the method proposed here presents an original form of data compression for grid DEMs, when combined with the linear-prediction algorithm. As such, the storage savings will be of benefit to all DEM users. In particular, the greatest benefits will be realised for users of multiple DEMs and microcomputer-users, for whom storage may be at a premium. The storage savings that are attained make it a favourable alternative to other grid DEM compression algorithms (Dutton, 1983; Shaffer, 1989). However, whilst these methods may offer similar compression ratios, elevation error is unconstrained, particularly in regions of variable terrain.

The results presented in Table 3 illustrate the significant data redundancy inherent within regular grid DEMs. This redundancy is due to the DEM being unadaptable to the variability of the terrain and the use of fixed-size storage units (ie. 16 bits). However, Huffman-encoding removes both these forms of data redundancy, the latter by the use of variable-length codes. The prediction algorithm assumes that there is a local linear trend within the data. Any deviations from this trend (ie. variable terrain) will be assigned longer, less probable codes, whilst uniform (ie. predictable) terrain will be assigned short codes. Hence, data redundancy is identified and eliminated at a local level.

Further DEM compression is possible with the introduction of small constrained elevation errors. This can be achieved by quantising or banding elevation classes together, in a similar manner to that of the Tiefert Mountain DEM. It was shown in Table 3 for this DEM, that the conversion of elevations from feet to metres, resulted in a further compression of average code length from 3.32 bits to 2.25 bits, or an increase in storage saving from 79.26% to 85.92%. This is accomplished with the introduction of absolute errors of less than 0.5 metres, due to rounding after conversion. Further results for ST06 and ST08 have shown that the introduction of similar constrained errors (ie. ± 0.5 metres), by rounding up or down all non sea level values (eg. elevation at one or two metres is represented by a

quantised value of 1.5 metres), produces a similar compression. This banding of elevations results in absolute average errors of 0.323 and 0.5 metres, respectively for ST06 and ST08, with corresponding average code lengths of 1.57 bits (-0.398) and 2.45 bits (-0.788) and storage savings of 90.19% (+2.49%) and 84.69% (+4.93%).

ACKNOWLEDGEMENTS

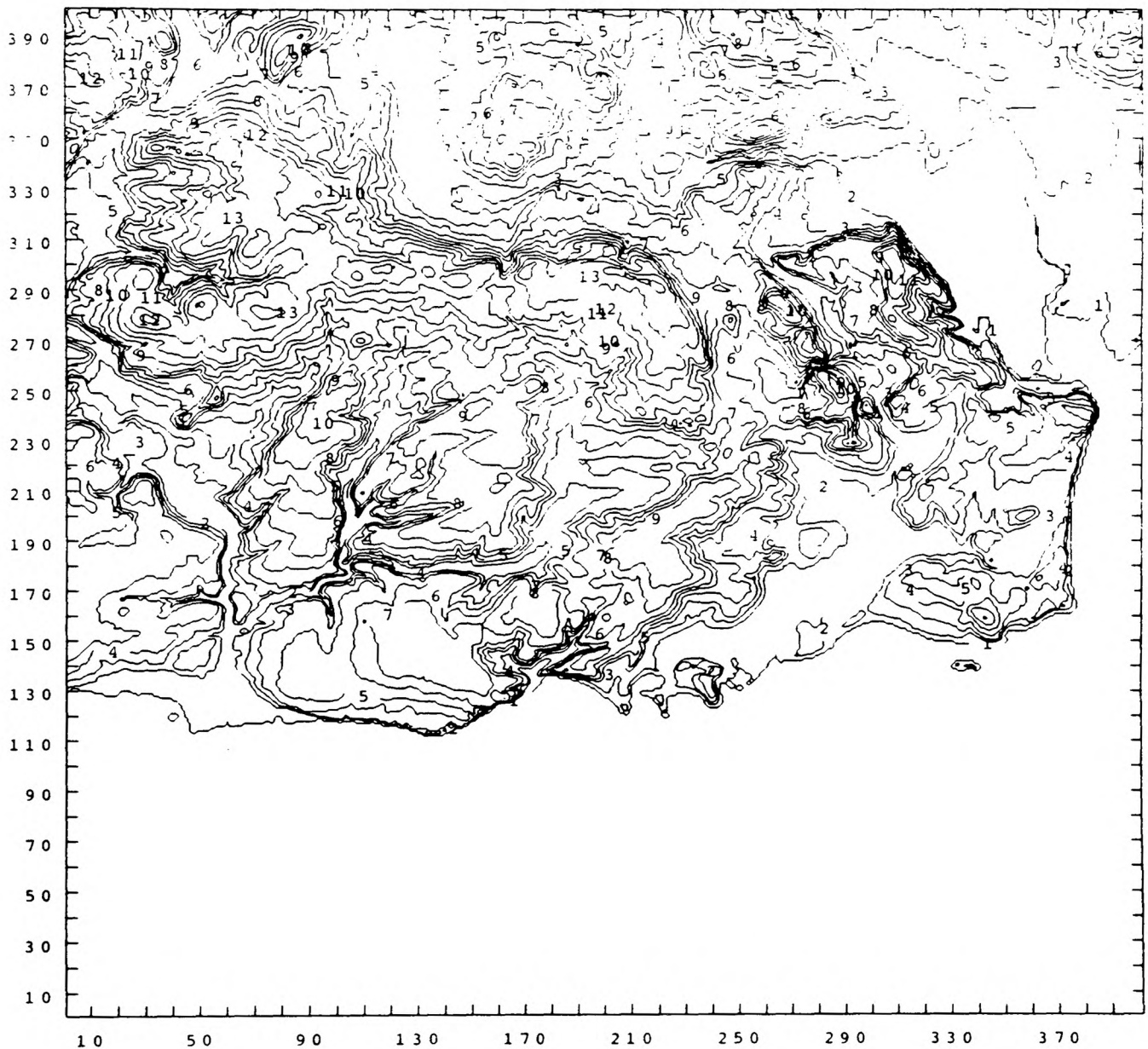
The authors would like to acknowledge the support given by the Royal Signals and Radar Establishment (R.S.R.E.), Ministry of Defence (Procurement Executive), Malvern, England. Thanks are also due to Mark Kumler of the U.S.G.S. and University of California, Santa Barbara for provision of the Tiefort Mountain DEM. The O.S. digital maps (ST06 and ST08) are reproduced with permission of the Controller of Her Majesty's Stationery Office (H.M.S.O.).

REFERENCES

- Ayeni, O.O. (1978) Automated digital terrain models: Proceedings of the Digital Terrain Models (DTM) Symposium, ASP/ACSM, St. Louis, Missouri, May 9-11, 1978, pp.275-306.
- Dutton, G. (1983) Efficient encoding of gridded surfaces: Spatial algorithms for processing land data with a microcomputer, Report of Organisation of the Lincoln Institute for Land Policy, Cambridge, Massachusetts, pp.23-62.
- Held, G. (1987) Data compression - techniques and applications, hardware and software considerations: John Wiley & Sons, G.B., 2nd edition, 206 pages.
- Huffman, D.A. (1952) A method for the construction of minimum-redundancy codes: Proceedings of IRE, Vol.40, Sept., pp.1098-1101. (Reprinted in Key papers in the development of information theory: edited by D. Slepian, IEE Press, New York, 1974, pp.47-50).
- Kidner, D.B., Jones, C.B. (1991) Implicit triangulations for large terrain databases: Proceedings of the Second European Conference on GIS (EGIS '91), Brussels, Belgium, Apr.2-5, Vol.1, pp.537-546.
- Lynch, T.J. (1985) Data compression - techniques and applications: Lifetime Learning Publications, Belmont, California, (Wadsworth), 345 pages.
- Petrie, G. (1987) Terrain data acquisition and modelling from existing maps, Proceedings of terrain modelling in Surveying and Civil Engineering, Univ. of Surrey, April 7-9, 1987, and Univ. of Glasgow, September 1-3, 1987.

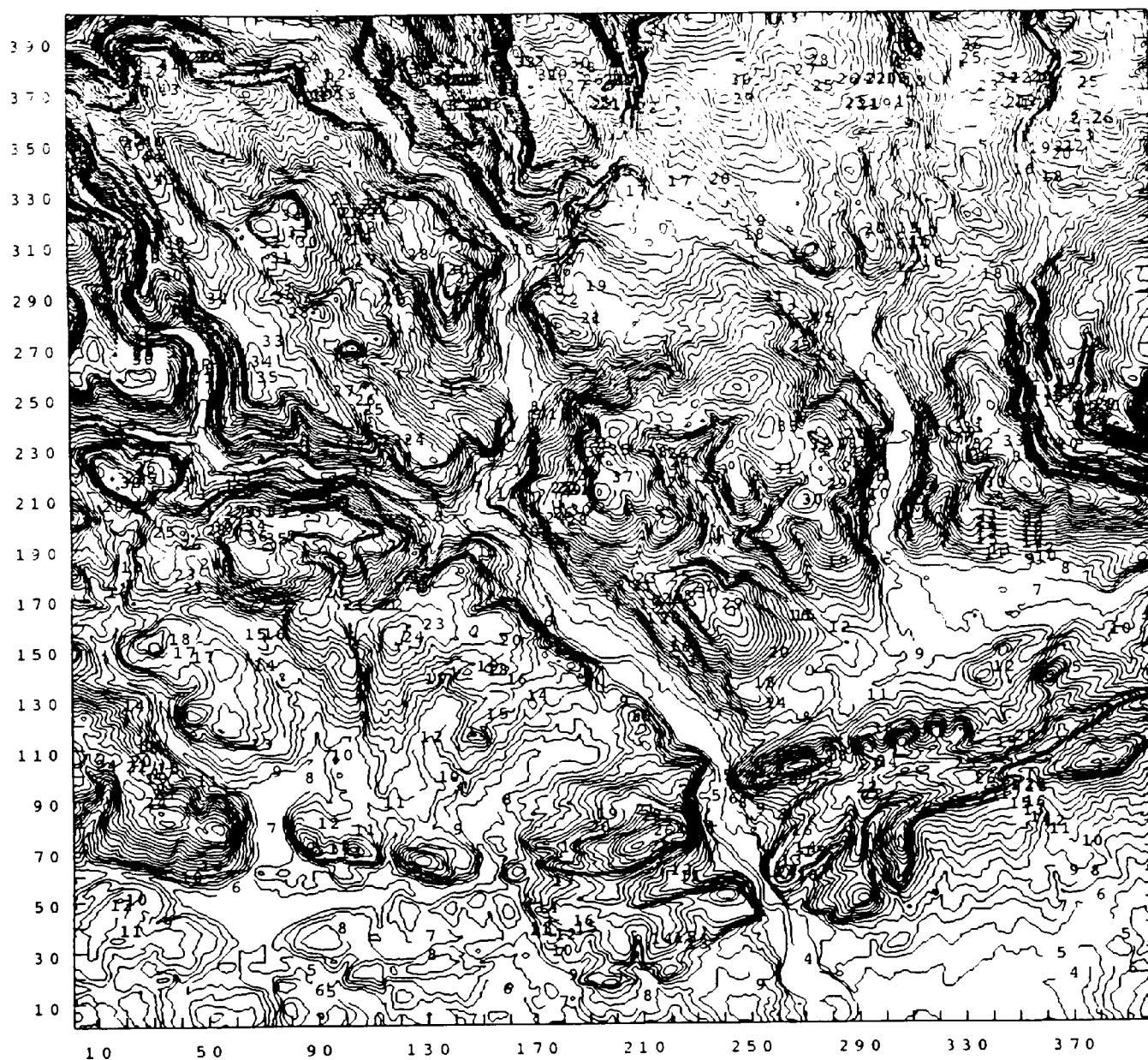
- Petrie, G., Kennie, T.J.M. (1987) An introduction to terrain modelling - applications and terminology: Proceedings of terrain modelling in Surveying and Civil Engineering, Univ. of Surrey, April 7-9, 1987, and Univ. of Glasgow, September 1-3, 1987.
- Shaffer, C.A. (1990) A full resolution elevation representation requiring three bits per pixel: In Design and Implementation of Large Spatial Databases, Edited by A.Buchmann, O. Günther, T.R. Smith and Y.-F. Wang, Springer Verlag, New York, pp. 45-64.
- United States Geological Survey (1990) Data users guide 5 - Digital elevation models: United States Department of the Interior, Reston, Virginia, 51 pages.
- Williams, C.M. (1986) The geometric modelling and compression of terrain data: Proceedings of the Second International Conference on Spatial Data Handling, Seattle, Washington, USA, July 5-10, pp.158-170.

APPENDIX A - REGULAR GRID DATA SETS



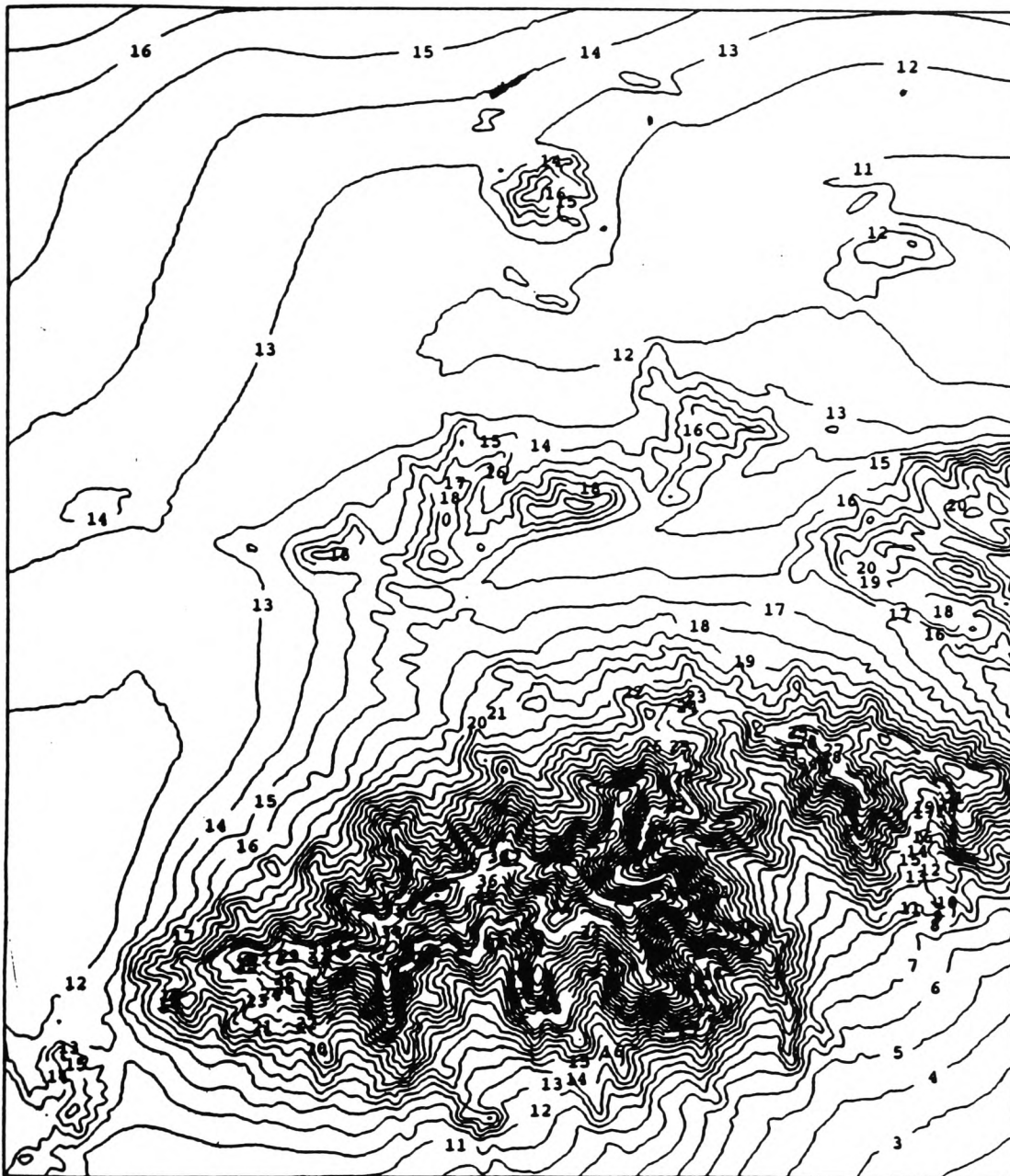
Contour Key (Elevations in Metres)		
1	0	11 100
2	10	12 110
3	20	13 120
4	30	14 130
5	40	15 140
6	50	
7	60	
8	70	
9	80	
10	90	

Figure 8. O.S. ST06 401x401 DEM of South Wales (Including Cardiff).



Contour Key (Elevations in Metres)					
1	0	11	100	21	200
2	10	12	110	22	210
3	20	13	120	23	220
4	30	14	130	24	230
5	40	15	140	25	240
6	50	16	150	26	250
7	60	17	160	27	260
8	70	18	170	28	270
9	80	19	180	29	280
10	90	20	190	30	290
				31	300
				32	310
				33	320
				34	330
				35	340
				36	350
				37	360
				38	370
				39	380
				40	390
				41	400
				42	410
				43	420
				44	430
				45	440
				46	450
				47	460
				48	470

Figure 9. O.S. ST08 401x401 DEM of the South Wales Valleys.



CONTOUR KEY	
1	450.0
2	475.0
3	500.0
4	525.0
5	550.0
6	575.0
7	600.0
8	625.0
9	650.0
10	675.0
11	700.0
12	725.0
13	750.0
14	775.0
15	800.0
16	825.0
17	850.0
18	875.0
19	900.0
20	925.0
21	950.0
22	975.0
23	1000.0
24	1025.0
25	1050.0
26	1075.0
27	1100.0
28	1125.0
29	1150.0
30	1175.0
31	1200.0
32	1225.0
33	1250.0
34	1275.0
35	1300.0
36	1325.0
37	1350.0
38	1375.0
39	1400.0
40	1425.0
41	1450.0
42	1475.0
43	1500.0
44	1525.0
45	1550.0

Figure 10. U.S.G.S. 376x461 DEM of the Tiefort Mountains, California.

APPENDIX B - PROGRAM DESCRIPTION & LISTING

The program is written in standard FORTRAN 77 and implemented on a DEC VAX 8800. There are four main subroutines in the program to calculate the Huffman codes (CALC_CODES) and look-up table (LOOK_UP), and to encode (ENCODE) and decode (DECODE) the grid DEM. Auxiliary subroutines include READ_DATA and WRITE_OUT_CODES which are self-explanatory, and MOVEBIT, SETBIT1 and GETBITS which perform the necessary bit operations. (In VAX FORTRAN, these bit operations can be performed with standard in-built library routines).

For simplicity, the program listing corresponds to the encoding of a standard O.S. 50 metre DEM, ie. 401x401 elevations, and the first row and column of the DEM are assumed to be known. It is left to the user to amend the program for DEMs of other resolutions and to difference encode the first row and column, if required.

Subroutine CALC_CODES

Before the Huffman codes can be calculated, a frequency distribution (array IFDIST) of values must be constructed. This corresponds to the corrections applied to the linearly predicted elevations. The construction of the codes is accomplished within a six-column work array IHUFF. These columns correspond to (1) the elevation correction value or code 'symbol'; (2) the frequency count; (3) the running sum of frequency count; (4) the codeword under construction; (5) the number of bits of the codeword; and (6) a pointer or marker linking codewords along similar paths of the binary tree.

Subroutine LOOK_UP

The list of codewords and their corresponding symbols or elevation corrections are constructed in the array CODES, while the look-up table for efficient search of the codewords is stored in the array POINTER. These fields correspond to the number of codewords of a certain bit-length and the start position in CODES for searching.

Subroutine ENCODE

From the list of codewords and array of corrections, the individual Huffman codes for each elevation within the DEM are packed into a 32-bit INTEGER array HUFFDEM, and referenced by ILONG.

Subroutine DECODE

Retrieval of the original grid DEM is accomplished in a two-stage process. Firstly, each grid node must be linearly-predicted and then its elevation difference must be decoded from the Huffman-encoded DEM, before the correction can be applied.

PROGRAM LISTING

```

PROGRAM HUFFMAN_CODING
C
C --- Program to calculate the Huffman codes for the linear predicted
C --- differences of a 401x401 Regular Grid DEM.
C
      INTEGER*2 ZCO(401,401),DIFFS(400,400),ICOUNT,ITEMPBITS(100)
      INTEGER*4 CODES(100,2),POINT(25,2)
      INTEGER*4 IHUFF(100,6),HUFFDEM(20000),ILONG
      COMMON /ELEVATIONS/ZCO
      COMMON /CORRECTIONS/DIFFS
      COMMON /WORKARRAY/IHUFF
      COMMON /SEARCH/CODES,POINT
      COMMON /COMPACT/ILONG,HUFFDEM
C
C --- ZCO is the 2-D array of original grid DEM elevations.
C --- DIFFS is the 2-D array of corrections, ie. values for coding.
C --- IHUFF is the work array used to construct the ICOUNT codes.
C --- CODES is the list of codes and their corresponding corrections.
C --- POINT is the look-up table into CODES, indicating no, and position.
C --- HUFFDEM is the compact DEM of Huffman codes.
C
      CALL READ_DATA
      CALL CALC_CODES(ICOUNT)
      CALL WRITE_OUT_CODES(ICOUNT)
      CALL LOOK_UP(ICOUNT,ITEMPBITS)
      CALL ENCODE(ICOUNT,ITEMPBITS)
      CALL DECODE(ICOUNT)
      END

      SUBROUTINE READ_DATA
      INTEGER*2 ZCO(401,401)
      CHARACTER FILENAME*8
      COMMON /ELEVATIONS/ZCO
      WRITE(6,10)
      READ(5,20) FILENAME
      OPEN(UNIT=10,NAME=FILENAME,STATUS='OLD')
      DO J=1,401
        DO K=1,20
          READ(10,30) (ZCO(I,J),I=(K-1)*20+1,K*20)
        END DO
        READ(10,40) ZCO(401,J)
      END DO
      CLOSE(UNIT=10)
10    FORMAT(/,' Input the File Name of the 401x401 Grid DEM ...')
20    FORMAT(A8)
30    FORMAT(20I4)
40    FORMAT(I4)
      RETURN
      END

      SUBROUTINE CALC_CODES(ICOUNT)
C
C --- Subroutine to linearly predict the grid of elevations and Huffman-encode
C --- the corrections (DIFFS). A frequency distribution of corrections is
C --- calculated (IFDIST), whilst the codes are constructed in the work-array
C --- IHUFF. The columns of which correspond to the elevation correction
C --- value; the frequency; the running sum of frequencies; the codeword under
C --- construction; the number of bits; and a marker linking codewords along
C --- similar paths of the binary tree.
C --- ICOUNT is the number of codewords.
C --- MINV1, MINV2, MINP1, MINP2 are the values and positions of the two least
C --- probabilities, whilst NEWMINV is their sum.
C --- MARKER is the current pointer to which all joined values (along the path
C --- of the binary tree) are set.
C --- ONE is a 1-bit binary value representing the code '1'.
C
      INTEGER*2 zco(401,401),DIFFS(400,400),ICOUNT
      INTEGER*4 IHUFF(100,6),IFDIST(-100:100)

```

```

INTEGER*4 MINV1,MINV2,MINP1,MINP2,NEWMINV,MARKER,ONE
COMMON /ELEVATIONS/ZCO
COMMON /CORRECTIONS/DIFFS
COMMON /WORKARRAY/IHUFF
DO I=-100,100                                ! Initialise arrays & variables
    IFDIST(I)=0
END DO
DO J=1,6
    DO I=1,100
        IHUFF(I,J)=0
    END DO
END DO
ICOUNT=0
MARKER=0
ONE=1
DO J=2,401                                    ! Predict elevations and form a
    DO I=2,401                                ! distribution of corrections.
        DIFFS(I-1,J-1)=ZCO(I,J)-ZCO(I-1,J)-ZCO(I,J-1)+ZCO(I-1,J-1)
        IFDIST(DIFFS(I-1,J-1))=IFDIST(DIFFS(I-1,J-1))+1
    END DO
END DO
DO I=-100,100                                ! Initialise Work Array
    IF (IFDIST(I).NE.0) THEN
        ICOUNT=ICOUNT+1
        IHUFF(ICOUNT,1)=I
        IHUFF(ICOUNT,2)=IFDIST(I)
        IHUFF(ICOUNT,3)=IFDIST(I)
    END IF
END DO

C
C --- Calculate the Huffman Codes.
C --- While the cumulated frequencies is not equal to the total number of
C --- points (160,000), combine the two lowest probabilities and replace with
C --- the joint probability; insert a '0' or '1' into their codes and for all
C --- other codes along the same path of the tree; and increment the number of
C --- bits.(All codes along a similar path of the tree are referenced with
C --- identical MARKER values).
C
DO WHILE (IHUFF(1,3).LT.160000)
    MARKER=MARKER+1
    MINV1=160000                                ! Find smallest probability
    DO I=1,ICOUNT
        IF (IHUFF(I,3).LT.MINV1) THEN
            MINV1=IHUFF(I,3)
            MINP1=I
        END IF
    END DO
    MINV2=160000                                ! Find next smallest with a
    DO I=1,ICOUNT                                ! different non-zero marker.
        IF (IHUFF(I,3).LT.MINV2.AND.I.NE.MINP1) THEN
            IF ((IHUFF(I,6).NE.IHUFF(MINP1,6)).OR.
                (IHUFF(I,6).EQ.0.AND.IHUFF(MINP1,6).EQ.0)) THEN
                MINV2=IHUFF(I,3)
                MINP2=I
            END IF
        END IF
    END DO
    NEWMINV=MINV1+MINV2
    IF (IHUFF(MINP1,6).NE.0) THEN                ! Insert '0' into codes
        DO I=1,ICOUNT
            IF (IHUFF(I,6).EQ.IHUFF(MINP1,6).AND.I.NE.MINP1) THEN
                IHUFF(I,3)=NEWMINV
                IHUFF(I,6)=MARKER
                IHUFF(I,5)=IHUFF(I,5)+1
            END IF
        END DO
    END IF
    IHUFF(MINP1,3)=NEWMINV
    IHUFF(MINP1,6)=MARKER
    IHUFF(MINP1,5)=IHUFF(MINP1,5)+1

```

```

      IF (IHUFF(MINP2,6).NE.0) THEN          ! Insert '1' into codes
        DO I=1,ICOUNT
          IF (IHUFF(I,6).EQ.IHUFF(MINP2,6).AND.I.NE.MINP2) THEN
            IHUFF(I,3)=NEWMINV
            IHUFF(I,6)=MARKER
            CALL MOVEBIT(ONE,0,IHUFF(I,4),IHUFF(I,5))
            IHUFF(I,5)=IHUFF(I,5)+1
          END IF
        END DO
      END IF
      IHUFF(MINP2,3)=NEWMINV
      IHUFF(MINP2,6)=MARKER
      CALL MOVEBIT(ONE,0,IHUFF(MINP2,4),IHUFF(MINP2,5))
      IHUFF(MINP2,5)=IHUFF(MINP2,5)+1
    END DO
  RETURN
END

```

```

SUBROUTINE WRITE_OUT_CODES(ICOUNT)
  INTEGER*2 ICOUNT
  INTEGER*4 IHUFF(100,6),K(32)
  REAL*4 SUMCODE,SUMENT,PROB
  COMMON /WORKARRAY/IHUFF
  SUMCODE=0.0
  SUMENT=0.0
  WRITE(6,10)
  DO I=1,ICOUNT
    DO J=1,IHUFF(I,5)
      K(J)=0
      CALL MOVEBIT(IHUFF(I,4),J-1,K(J),0)
    END DO
    PROB=FLOAT(IHUFF(I,2))/160000.0
    SUMCODE=SUMCODE+PROB*IHUFF(I,5)
    SUMENT=SUMENT-PROB*(LOG10(PROB)/LOG10(2.0))
    WRITE(6,20) IHUFF(I,1),IHUFF(I,2),PROB,IHUFF(I,4),IHUFF(I,5),
      * (K(L),L=IHUFF(I,5),1,-1)
  END DO
  WRITE(6,30) SUMCODE,SUMENT,(SUMENT/SUMCODE)*100.0,100.0-(SUMCODE/0.16)
10  FORMAT(//,' Height',23X,'Integer No.of',
  * -//,' Diff. Freq. Probability Code Bits Code',
  * //,66('='))
20  FORMAT(I6,I8,F14.9,I8,I6,4X,20I1)
30  FORMAT(//,' The Average Code Length =',F8.5,' bits/elevation',
  * //,' The Entropy =',F8.5,' bits/elevation',
  * //,' The Code Efficiency =',F8.4,'% ',
  * //,' The Storage Saving =',F8.4,'% ',/)
  RETURN
END

```

```

SUBROUTINE LOOK_UP(ICOUNT,ITEMPBITS)
C
C --- Formulate the sorted list of codewords (CODES) in ascending bit-length
C --- (ie. decreasing probability) and a look-up table (POINT) for efficient
C --- search of CODES, since the most probable values are searched first.
C
  INTEGER*2 ICOUNT, CODECOUNT, ITEMPBITS(100)
  INTEGER*4 CODES(100,2), POINT(25,2), IHUFF(100,6)
  COMMON /WORKARRAY/IHUFF
  COMMON /SEARCH/CODES, POINT
  DO J=1,2
    DO I=1,100
      CODES(I,J)=0
    END DO
  END DO
  DO I=1,100
    ITEMPBITS(I)=0
  END DO
  DO I=1,25

```

```

        POINT(I,1)=0
        POINT(I,2)=0
    END DO
C
C --- Search for the codeword of greatest frequency, each time.
C
    IPTR=0
    IBIT=1
    CODECOUNT=0
    DO WHILE(CODECOUNT.NE.ICOUNT)
        IFREQ=0
        ICODE=0
        IDIFF=0
        IPOSN=0
        DO I=1,ICOUNT
            IF (IHUFF(I,5).EQ.IBIT.AND.IHUFF(I,2).GT.IFREQ
                .AND.IHUFF(I,6).NE.0) THEN
                IFREQ=IHUFF(I,2)
                ICODE=IHUFF(I,4)
                IDIFF=IHUFF(I,1)
                IPOSN=I
            END IF
        END DO
        IF (IPOSN.EQ.0) THEN
            POINT(IBIT,1)=IPTR
            IF (IBIT.EQ.1) POINT(IBIT,2)=1
            IF (IBIT.NE.1) POINT(IBIT,2)=POINT(IBIT-1,2)+POINT(IBIT-1,1)
            IBIT=IBIT+1
            IPTR=0
        ELSE
            IPTR=IPTR+1
            CODECOUNT=CODECOUNT+1
            CODES(CODECOUNT,1)=ICODE
            CODES(CODECOUNT,2)=IDIFF
            ITEMPSBITS(CODECOUNT)=IBIT
            IHUFF(IPOSN,6)=0
        END IF
    END DO
    POINT(IBIT,1)=IPTR
    POINT(IBIT,2)=POINT(IBIT-1,2)+POINT(IBIT-1,1)
    WRITE(6,10)
    DO I=1,IBIT
        IF (POINT(I,1).NE.0) WRITE(6,20) I,POINT(I,1),POINT(I,2)
    END DO
    WRITE(6,30)
    DO I=1,ICOUNT
        WRITE(6,40) I,CODES(I,1),CODES(I,2)
    END DO
10  FORMAT(//,' ----- Look-Up Table -----',
        *      /,' Bits Frequency Position',/,26('='))
20  FORMAT(I4,I9,I9)
30  FORMAT(//,' ----- CodeWord Table -----',
        *      /,' Position Code Correction',/,29('='))
40  FORMAT(I6,2I9)
    RETURN
    END

SUBROUTINE ENCODE(ICOUNT,ITEMPSBITS)
C
C --- Subroutine to encode the differences between the original and predicted
C --- values (DIFFS) into strings of Huffman codes, stored (packed) into an
C --- array of 32-bit INTEGERS (HUFFDEM). The current longword value is
C --- referenced as ILONG, whilst the current bit is IBIT.
C
    INTEGER*2 DIFFS(400,400),ICOUNT,ITEMPSBITS(100)
    INTEGER*4 CODES(100,2),POINT(25,2),HUFFDEM(20000)
    INTEGER*4 ILONG,IBIT,K(32)
    COMMON /CORRECTIONS/DIFFS
    COMMON /SEARCH/CODES,POINT ~

```

```

COMMON      /COMPACT/ILONG,HUFFDEM
IBIT=31
ILONG=1
HUFFDEM(ILONG)=0
DO J=1,400
  DO I=1,400
    INDEX=1
    DO WHILE(DIFFS(I,J).NE.CODES(INDEX,2))
      INDEX=INDEX+1
    END DO
    ICODE=CODES(INDEX,1)
    NBITS=ITEMPBITS(INDEX)
    DO M=1,NBITS
      K(M)=0
      CALL MOVEBIT(ICODE,M-1,K(M),0)
    END DO
    DO L=NBITS,1,-1
      IF(K(L).EQ.1) CALL SETBIT1(HUFFDEM(ILONG),IBIT)
      IBIT=IBIT-1
      IF (IBIT.EQ.-1) THEN
        IBIT=31
        ILONG=ILONG+1
        HUFFDEM(ILONG)=0
      END IF
    END DO
  END DO
END DO
WRITE(6,10) ILONG
10  FORMAT(/,I6,' 32-bit Longwords Encoded',/)
RETURN
END

SUBROUTINE DECODE(ICOUNT)
c
c --- Subroutine to decode the Huffman code for each difference between the
c --- predicted and original elevation. This correction is applied to the
c --- linearly predicted value to reform the original DEM (NEWZ). The first
c --- row and column are assumed to be known, but could also be coded as the
c --- differences between elevations, using the same set of codes.
c --- GETBITS returns the 32 individual bits of the current longword (HUFFDEM)
c
  INTEGER*2 ZCO(401,401),NEWZ(401,401),ICOUNT
  INTEGER*4 CODES(100,2),POINT(25,2),HUFFDEM(20000)
  INTEGER*4 ILONG,IBIT,ITOTAL,IBITVALUE,K(0:31)
  COMMON      /ELEVATIONS/ZCO
  COMMON      /SEARCH/CODES,POINT
  COMMON      /COMPACT/ILONG,HUFFDEM
  DO I=1,401
    NEWZ(I,1)=ZCO(I,1)
    NEWZ(1,I)=ZCO(1,I)
  END DO
  IBIT=31
  ILONG=1
  CALL GETBITS(HUFFDEM(ILONG),K)
  DO J=2,401
    DO I=2,401
      IMATCH=0
      ICURRCODE=0
      NBITS=0
      DO WHILE (IMATCH.EQ.0)
        IBITVALUE=K(IBIT)
        ICURRCODE=ICURRCODE*2+IBITVALUE
        NBITS=NBITS+1
        IBIT=IBIT-1
        IF (IBIT.EQ.-1) THEN
          IBIT=31
          ILONG=ILONG+1
          CALL GETBITS(HUFFDEM(ILONG),K)
        END IF
      END WHILE
    END DO
  END DO
  ! Copy first row and column

```

```

      IF (POINT(NBITS,1).NE.0) THEN
        ISTARTS=POINT(NBITS,2)
        IFINSHS=ISTARTS+POINT(NBITS,1)-1
        DO WHILE(ISTARTS.LE.IFINSHS.AND.IMATCH.EQ.0)
          IF (CODES(ISTARTS,1).EQ.ICURRCODE) THEN
            IMATCH=1
            NEWZ(I,J)=CODES(ISTARTS,2)+NEWZ(I-1,J)
              +NEWZ(I,J-1)-NEWZ(I-1,J-1)
          END IF
          ISTARTS=ISTARTS+1
        END DO
      END IF
    END DO
  END DO
END DO
WRITE(6,10) ILONG
10  FORMAT(/,I6,' 32-bit Longwords Decoded',/)
RETURN
END

SUBROUTINE MOVEBIT(IWORD,IPOS,JWORD,JPOS)
C
C --- Move the bit in position IPOS of IWORD to position JPOS in JWORD.
C
  INTEGER*4 IWORD,IPOS,JWORD,JPOS,ITEMP
  ITEMP=IWORD
  DO I=0,IPOS-1
    ITEMP=ITEMP/2
  END DO
  IVALUE=ITEMP-(ITEMP/2)*2
  IF (IVALUE.EQ.1) JWORD=JWORD+2**JPOS
  RETURN
END

SUBROUTINE SETBIT1(JWORD,JPOS)
C
C --- Set the bit in position JPOS of JWORD.
C
  PARAMETER (IFIRSTBIT=-2147483648)
  INTEGER*4 JWORD,JPOS
  IF (JPOS.NE.31) JWORD=JWORD+2**JPOS
  IF (JPOS.EQ.31) JWORD=JWORD+IFIRSTBIT
  RETURN
END

SUBROUTINE GETBITS(LWORD,K)
C
C --- Return the 32 individual bits (K) of a longword (LWORD).
C --- (N.B. Two's complement a negative value).
C
  INTEGER*4 LWORD,ITEMP,K(0:31)
  ITEMP=ABS(LWORD)
  DO I=0,31
    K(I)=MOD(ITEMP,2)
    ITEMP=ITEMP/2
  END DO
  IF (LWORD.LT.0) THEN
    I=0
    DO WHILE(K(I).EQ.0)
      I=I+1
    END DO
    DO J=I+1,31
      IF (K(J).EQ.0) THEN
        K(J)=1
      ELSE
        K(J)=0
      END IF
    END DO
  END IF
  RETURN
END

```


APPENDIX C - EXAMPLE OUTPUT

```
%PHOTO-I-INIT, Recording initiated at 12-JUL-1991 11:15:46.09
$
$ run huffman
```

```
Input the File Name of the 401x401 Grid DEM ...
st06.dat
```

Height Diff.	Freq.	Probability	Integer Code	No.of Bits	Code
-40	1	0.000006250	79406	18	010011011000101110
-33	1	0.000006250	79407	18	010011011000101111
-29	1	0.000006250	81016	18	010011110001111000
-26	4	0.000025000	20255	16	0100111100011111
-22	2	0.000012500	40509	17	01001111000111101
-21	1	0.000006250	81017	18	010011110001111001
-20	7	0.000043750	10125	15	010011110001101
-19	3	0.000018750	20240	16	0100111100010000
-18	3	0.000018750	20241	16	0100111100010001
-17	7	0.000043750	10126	15	010011110001110
-16	6	0.000037500	10121	15	010011110001001
-15	6	0.000037500	10124	15	010011110001100
-14	10	0.000062500	4989	14	01001101111101
-13	12	0.000075000	5061	14	01001111000101
-12	34	0.000212500	1241	12	010011011001
-11	32	0.000200000	2539	13	0100111101011
-10	59	0.000368750	1268	12	010011110100
-9	75	0.000468750	622	11	01001101110
-8	109	0.000681250	633	11	01001111001
-7	194	0.001212500	315	10	0100111011
-6	286	0.001787500	154	9	010011010
-5	473	0.002956250	159	9	010011111
-4	965	0.006031250	33	7	0100001
-3	2067	0.012918750	18	6	010010
-2	5189	0.032431249	11	5	01011
-1	19253	0.120331250	3	3	011
0	102277	0.639231265	1	1	1
1	19427	0.121418752	0	2	00
2	5146	0.032162499	10	5	01010
3	2035	0.012718750	17	6	010001
4	951	0.005943750	32	7	0100000
5	523	0.003268750	76	8	01001100
6	301	0.001881250	156	9	010011100
7	161	0.001006250	314	10	0100111010
8	128	0.000800000	635	11	01001111011
9	74	0.000462500	621	11	01001101101
10	45	0.000281250	1264	12	010011110000
11	39	0.000243750	1246	12	010011011110
12	29	0.000181250	2538	13	0100111101010
13	23	0.000143750	2495	13	0100110111111
14	9	0.000056250	4963	14	01001101100011
15	8	0.000050000	4960	14	01001101100000
16	4	0.000025000	9922	15	010011011000010
17	9	0.000056250	4988	14	01001101111100
18	2	0.000012500	19846	16	0100110110000110
19	2	0.000012500	19847	16	0100110110000111
20	1	0.000006250	39696	17	01001101100010000
21	1	0.000006250	39697	17	01001101100010001
23	1	0.000006250	39698	17	01001101100010010
25	1	0.000006250	39699	17	01001101100010011
26	2	0.000012500	19850	16	0100110110001010
41	1	0.000006250	39702	17	01001101100010110

```
The Average Code Length = 1.96809 bits/elevation
The Entropy              = 1.88104 bits/elevation
The Code Efficiency      = 95.5772%
The Storage Saving       = 87.6995%
```

----- Look-Up Table -----
Bits Frequency Position

1	1	1
2	1	2
3	1	3
5	2	4
6	2	6
7	2	8
8	1	10
9	3	11
10	2	14
11	4	16
12	4	20
13	3	24
14	5	27
15	5	32
16	6	37
17	6	43
18	4	49

----- CodeWord Table -----
Position Code Correction

1	1	0
2	0	1
3	3	-1
4	11	-2
5	10	2
6	18	-3
7	17	3
8	33	-4
9	32	4
10	76	5
11	159	-5
12	156	6
13	154	-6
14	315	-7
15	314	7
16	635	8
17	633	-8
18	622	-9
19	621	9
20	1268	-10
21	1264	10
22	1246	11
23	1241	-12
24	2539	-11
25	2538	12
26	2495	13
27	5061	-13
28	4989	-14
29	4963	14
30	4988	17
31	4960	15
32	10125	-20
33	10126	-17
34	10121	-16
35	10124	-15
36	9922	16
37	20255	-26
38	20240	-19
39	20241	-18
40	19846	18
41	19847	19
42	19850	26
43	40509	-22
44	39696	20
45	39697	21
46	39698	23
47	39699	25
48	39702	41
49	79406	-40
50	79407	-33
51	81016	-29
52	81017	-21

9841 32-bit Longwords Encoded

9841 32-bit Longwords Decoded

\$ photo/off

Supplement - Paper Four

%PHOTO-I-INIT, Recording initiated at 12-JUL-1991 11:16:30.75

\$

\$ run huffman

Input the File Name of the 401x401 Grid DEM ...
st08.dat

Height Diff.	Freq.	Probability	Integer Code	No.of Bits	Code
-39	1	0.000006250	41932	17	01010001111001100
-32	2	0.000012500	20967	16	0101000111100111
-31	2	0.000012500	26900	16	0110100100010100
-28	1	0.000006250	41933	17	01010001111001101
-27	3	0.000018750	10482	15	010100011110010
-26	1	0.000006250	53802	17	01101001000101010
-24	1	0.000006250	53803	17	01101001000101011
-21	1	0.000006250	53804	17	01101001000101100
-20	9	0.000056250	6727	14	01101001000111
-19	14	0.000087500	3360	13	0110100100000
-18	20	0.000125000	1294	12	010100001110
-17	12	0.000075000	2619	13	0101000111011
-16	41	0.000256250	652	11	01010001100
-15	42	0.000262500	653	11	01010001101
-14	68	0.000425000	320	10	0101000000
-13	69	0.000431250	321	10	0101000001
-12	106	0.000662500	417	10	0110100001
-11	125	0.000781250	421	10	0110100101
-10	250	0.001562500	211	9	011010011
-9	375	0.002343750	83	8	01010011
-8	553	0.003456250	106	8	01101010
-7	829	0.005181250	42	7	0101010
-6	1332	0.008325000	58	7	0111010
-5	2119	0.013243750	27	6	011011
-4	3650	0.022812501	11	5	01011
-3	6654	0.041587502	4	4	0100
-2	13201	0.082506247	13	4	1101
-1	28571	0.178568751	0	2	00
0	45104	0.281899989	2	2	10
1	27481	0.171756253	7	3	111
2	12870	0.080437496	12	4	1100
3	6642	0.041512500	15	5	01111
4	3673	0.022956250	12	5	01100
5	2208	0.013800000	28	6	011100
6	1338	0.008362500	59	7	0111011
7	906	0.005662500	43	7	0101011
8	555	0.003468750	107	8	01101011
9	353	0.002206250	82	8	01010010
10	234	0.001462500	209	9	011010001
11	163	0.001018750	162	9	010100010
12	105	0.000656250	416	10	0110100000
13	80	0.000500000	322	10	0101000010
14	64	0.000400000	841	11	01101001001
15	40	0.000250000	646	11	01010000110
16	27	0.000168750	1311	12	010100011111
17	22	0.000137500	1308	12	010100011100
18	21	0.000131250	1295	12	010100001111
19	13	0.000081250	2621	13	0101000111101
20	14	0.000087500	3361	13	0110100100001
21	11	0.000068750	2618	13	0101000111010
22	7	0.000043750	6724	14	01101001000100
23	6	0.000037500	5240	14	01010001111000
24	2	0.000012500	26903	16	0110100100010111
26	2	0.000012500	26904	16	0110100100011000
27	2	0.000012500	26905	16	0110100100011001
31	2	0.000012500	26906	16	0110100100011010
35	1	0.000006250	53805	17	01101001000101101
46	1	0.000006250	53814	17	01101001000110110
49	1	0.000006250	53815	17	01101001000110111

The Average Code Length = 3.23754 bits/elevation
The Entropy = 3.18300 bits/elevation
The Code Efficiency = 98.3154%
The Storage Saving = 79.7654%

----- Look-Up Table -----
 Bits Frequency Position

2	2	1
3	1	3
4	3	4
5	3	7
6	2	10
7	4	12
8	4	16
9	3	20
10	6	23
11	4	29
12	4	33
13	5	37
14	3	42
15	1	45
16	6	46
17	8	52

----- CodeWord Table -----
 Position Code, Correction

1	2	0
2	0	-1
3	7	1
4	13	-2
5	12	2
6	4	-3
7	15	3
8	12	4
9	11	-4
10	28	5
11	27	-5
12	59	6
13	58	-6
14	43	7
15	42	-7
16	107	8
17	106	-8
18	83	-9
19	82	9
20	211	-10
21	209	10
22	162	11
23	421	-11
24	417	-12
25	416	12
26	322	13
27	321	-13
28	320	-14
29	841	14
30	653	-15
31	652	-16
32	646	15
33	1311	16
34	1308	17
35	1295	18
36	1294	-18
37	3360	-19
38	3361	20
39	2621	19
40	2619	-17
41	2618	21
42	6727	-20
43	6724	22
44	5240	23
45	10482	-27
46	20967	-32
47	26900	-31
48	26903	24
49	26904	26
50	26905	27
51	26906	31
52	41932	-39
53	41933	-28
54	53802	-26
55	53803	-24
56	53804	-21
57	53805	35
58	53814	46
59	53815	49

16188 32-bit Longwords Encoded

16188 32-bit Longwords Decoded

\$ photo/off